AFIT/GOR/ENC/95M-01

Estimation of the Captive-Carry Survival Function for the Advanced Medium Range Air-to-Air Missile (AMRAAM)

THESIS

David R. Denhard
Captain, USAF

AFIT/GOR/ENC/95M-01

AFIT/GOR/ENC/95M-01

Estimation of the Captive-Carry Survival Function for the Advanced Medium Range Air-to-Air Missile (AMRAAM)

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Operations Research

David R. Denhard, B.S. in Mechanical Engineering
Captain, USAF

March, 1995

Approved for public release; distribution unlimited.

Acknowledgements

I would like to express my sincere thanks to Major Kevin Lawson and Lieutenant Colonel Dennis Dietz for introducing me to this research opportunity and guiding me through the thesis process. I wish to give a special thanks to Mr Paul Giglielmoni and Major William Kobren for their patience in answering questions about the AMRAAM program and to Dr Hani Doss for providing the random number generators used in the Fortran program implementing Lawson's non-parametric Bayesian estimator.

David R. Denhard

Table of Contents

Page

List of Figures

List of Tables

Table		Page

AFIT/GOR/ENC/95M-01

Abstract

This thesis considers the problem of estimating the survival function of an item (probability that the item functions for a time greater than a given time t) from sampling data subject to partial right censoring (a portion of the items in the sampling data have not yet been observed to fail). Specifically, the thesis describes several parametric and non-parametric statistical models that can be used when the sampling data is subject to partial right censoring. These models are applied to the case of estimating the captive-carry survival function of the AIM-120A Advanced Medium Range Air-to-Air Missile (AMRAAM). The captive-carry survival function for the AMRAAM exhibits regions of exponential behavior (i.e., constant failure rate), but the survival function is not entirely exponential. The non-parametric statistical models indicate the regions of increasing /decreasing failure rate in the AMRAAM captive-carry survival function and provide a robust set of investigative tools for estimating the survival function of any item.

Estimation of the Captive-Carry Survival Function for the Advanced Medium Range Air-to-Air Missile (AMRAAM)

I. Introduction

The thesis considers the problem of estimating the survival function of an item (probability that the item functions for a time greater than a given time t) from sampling data subject to partial right censoring (a portion of the items in the sampling data have not yet been observed to fail). The thesis is written in a *case study* format. Specifically, the case of estimating the captive-carry survival function of the AIM-120A Advanced Medium Range Air-to-Air Missile (AMRAAM) is considered. Chapter 1 introduces the thesis effort, providing an overview of the problem of estimating the captive-carry survival function for the AMRAAM. Chapter 2 (Literature Review) provides a detailed description of the statistical models that be utilized to estimate the survival function of an item when sampling data is subject to partial right censoring. Chapter 3 (Methodology) provides algorithms for implementing these statistical models. Chapter 4 (Findings) documents the results of the statistical analysis of the captive-carry survival function for the AMRAAM. Chapter 5 (Conclusion) outlines the conclusions that may be drawn from the statistical analysis and provides a taxonomy for estimating the survival function of an item from sampling data subject to partial right censoring.

This chapter contains four sections. Section 1.1 (Background) begins with a brief description of the AMRAAM, continues with an introduction of the reliability metric: *captive-carry lifelength*, and ends with a description of how military personnel assess the operational status of the missile. Section 1.2 (Problem Description) defines the thesis problem; details the current approach to solving the problem; and introduces the factors that obfuscate solution(s) to the problem. Section 1.3 (Scope) addresses the issue of estimating the captive-carry survival function from the system versus component level. This section also introduces the statistical models used in the estimation.

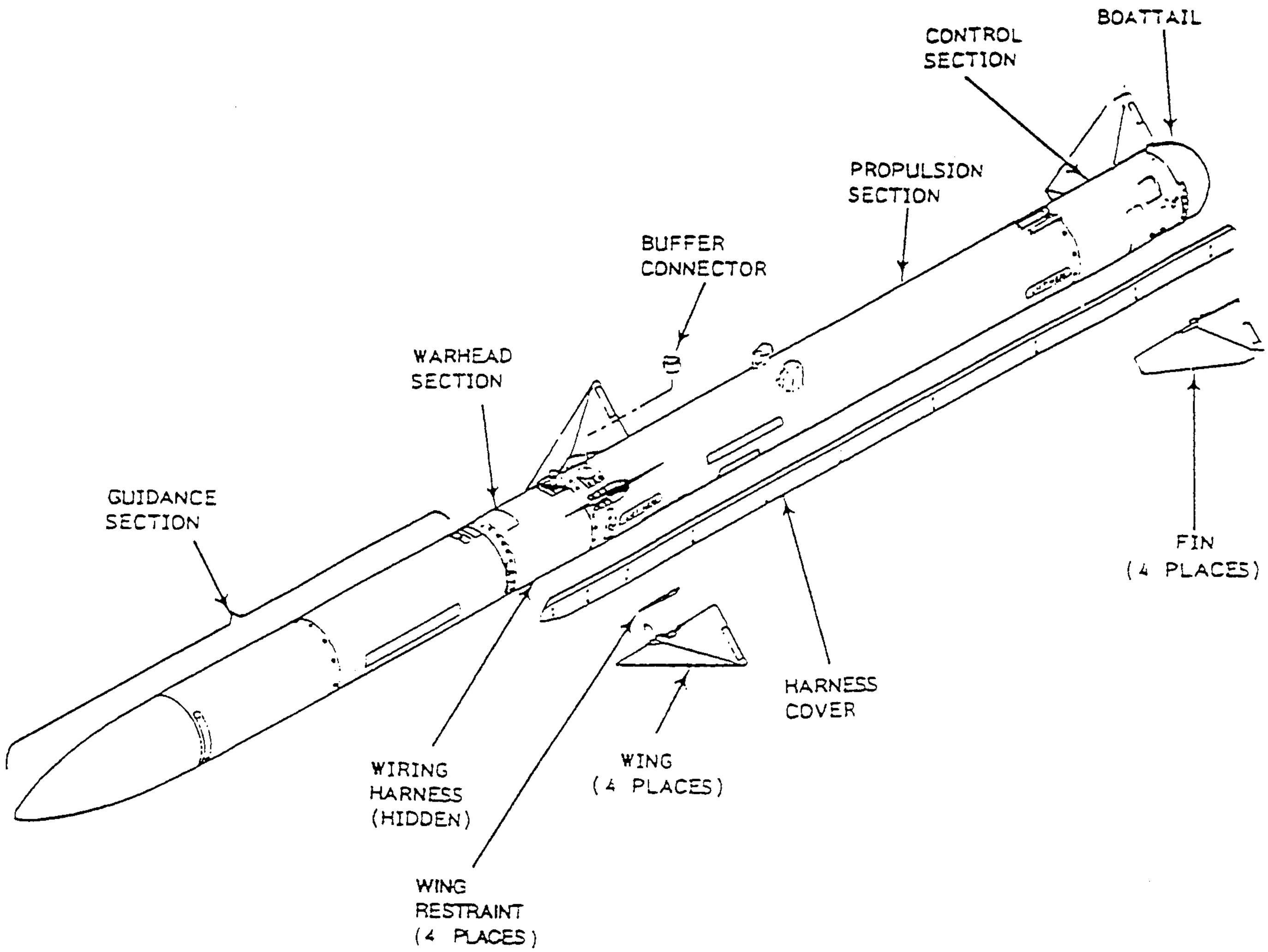


Figure 1. Schematic Diagram of AMRAAM

Section 1.4 provides a summary of the chapter and an outline of the material to be covered in succeeding chapters.

1.1 Background

1.1.1 AMRAAM. The AIM-120A, AMRAAM, provides United States Air Force (USAF) F-15 and F-16 fighter aircraft and United States Navy (USN) F-14 and F-18 fighter aircraft with a *launch and leave* fighting capability. Specifically, the AMRAAM's active radar terminal homing seeker enables the missile, when fired, to *lock on* a target without further assistance from the launching aircraft. The AMRAAM replaces the radar guided AIM-7 which requires the launching

aircraft to provide constant radar illumination in order to deliver the AIM-7 on target. In contrast, for a typical launch profile, the AMRAAM uses the launching aircraft's radar for initial and mid-course guidance and then switches to its own radar for active homing through intercept.

An AMRAAM consists of four functional sections: guidance, armament, propulsion, and control (Figure 1 (HMSC, Jan 94)). The guidance section performs functions necessary for midcourse guidance and control, target acquisition, terminal guidance and control, and target encounter timing for warhead detonation. The guidance section contains the antenna, the seeker assembly, and the missile electronics. The armament section contains a blast fragmentation warhead and a safe and arm device which detonates the warhead. The propulsion section contains the rocket motor, the hooks used for both the rail and ejection launcher, the flush-mounted missile umbilical, and the sockets for the detachable wings. The buffer connector (top, forward of the propulsion section) is the electrical interface between the missile and the launcher. The control section contains the complete control actuation system including electronics and four control surfaces.

1.1.2 Captive-Carry Lifelength. Captive-carry lifelength measures the cumulative time a missile can be carried by an aircraft **in-flight** and still remain launch capable. Accurate estimation of the distribution of captive-carry lifelength or captive-carry survival function is crucial as the AMRAAM operational concept consists of captive-carrying a missile until a failure is detected. The military services required the AMRAAM be designed for a captive-carry mean time between failures (MTBF) of 1000 hours. However, in early production testing, the missile demonstrated a low frequency vibration problem when captive-carried under the fuselage of an F-15 aircraft. The August 1990 issue of International Defense Review reports:

> *Missiles mounted on the F-15's fuselage stations fell short of the Air Force's reliability target of 220 [hours] during captive-carriage trials, as excessive vibration caused a series of mechanical failures. When the pilot of an F-15 throttles back during maneuvering – such as in a wind-up turn – air that would normally enter the engine intakes may instead spill over the fuselage mounted AMRAAMs. This induces vibrations at 200-300 Hz, in which range the missile's electronics are vulnerable. (Hewishi, Robinson, and Tubé (1990))*

In the aforementioned F-15 trial, missile damage included severe control fin bending (as extreme as 180 degrees) and unzipping of electronic cards from their host boards. As a result of the low frequency vibration problem, the government revised the captive-carry MTBF *acceptance* requirement from 1000 hours to 450 hours[1] (Guglielmoni, 1994).

Engineers designed the AMRAAM to be captive-carried dormant; power does not have to be continually supplied to the missile during flight. As a result of the military services' requirement for a small missile cross section (seven inches), designers housed the missile's electronic components close together. The dormant captive-carry requirement prevents the electronic components from overheating.

1.1.3 Assessing AMRAAM Captive-Carry Lifelength. In order to assess the operational readiness of the missile in a minimal amount of *power on* time, designers created an automated, three second built-in test (BIT) (Guglielmoni, 1994)[2]. An aircraft's crew or ground personnel, either in flight or on the ground, can perform a BIT given external power and data bus messages supplied by an aircraft[3]. The AMRAAM data processor (ADP), housed in the electronics compartment of the guidance section, controls the BIT and supplies status information back to the aircraft's central computer for evaluation (HMSC, 1994).

BIT assesses the status of the missile with a "86 percent thoroughness" (Guglielmoni, 1994). Table 1 shows a break out of BIT effectiveness , "the portion of the assembly exercised during performance of BIT as determined by engineering analysis,", for the section, unit, and assembly levels (HMSC, 1994). For instance, referring to Table 1, BIT tests the Antenna assembly of the Antenna-Receiver/Transmitter unit (located in guidance section) with a bit effectiveness of 95 percent. BIT outputs a set of degraded mode assessment(DMA) and BIT status messages that

[1]Raytheon company, one of the missile's manufacturers, has agreed to an acceptance requirement of 650 hours for Production Lot 6 missiles.

[2]BIT can be repeated up to ten times within a ten minute period without causing damage to the missile.

[3]BIT may also be performed off-line with a Missile BIT Test Set or "Suit Case" Tester.

Section / Unit / Assembly	Bit Effectiveness
1. Guidance Section	
a) Antenna-Receiver/Transmitter	
1) Antenna	.95
2) Transmitter/Electronic Conversion Unit	.25
3) Battery and Cables	NC
b) Remote Terminal	.95
c) Program Memory	.60
d) Launch Sequencer	.60
e) Input/Output	.85
f) ADP and Operand Memory	.90
g) Filter Processor	.90
h) Receiver/Range Correlator	.85
i) Frequency Reference Unit	.85
j) Inertial Reference Unit	.85
k) Target Detection Device	
1. Radio Frequency Head	.25
2. Video	.95
3. Cables	NC
l) Backplane Assembly	.95
m) Forward, Aft Fuselage and Misc. Cables	NC
2. Armament Section	NC
3. Propulsion Section	NC
4. Control Section	
a) Electrical	.70
b) Mechanical	.75
5. Auxiliary	
a) Data Link Receiver	.85
b) Rectifier/Filter	.80
c) Main Wiring	.90

NC - not checked by BIT

Table 1. BIT Effectiveness for the Section, Unit, and Assembly Levels

report, respectively, the aggregate results of the BIT and specific BIT unit tests. The DMA messages are:

1. Failed BIT
2. Guidance Failure
3. Transmitter Failure
4. Data Link Failure
5. Failure - No Mission Impact
6. Passed BIT
7. No Missile-1553B
8. No Test.

In addition to BIT, a Fully Automated System Test (FAST) may be performed using an off-line facility or mission test station. FAST consists of a BIT, a launch cycle test (LCT), and a hardware verification test. The FAST BIT is a five second test identical to the 3 second BIT with the addition of an external data link test. The LCT is a six second test of launch cycle functions including power changeover, launch cycle event timing, navigation initialization, two-way bus communication, antenna control, and commit to launch. The hardware verification test is a 47 second test of the missile hardware; the test measures over 700 specific parameters. FAST is a "more comprehensive functional test than [three second] BIT; a successful BIT does not imply a successful FAST". (NAWC, 1992). However, FAST is a facility only test and can not performed in the field.

1.2 Problem Description

The Air-to-Air Joint System Program Office (JSPO) AMRAAM Integrated Product Team (IPT) and Headquarters Air Force Operational Test and Evaluation Center (HQ AFOTEC) request

the statistical modeling of the captive-carry survival function for the AMRAAM. Specifically, these organizations request:

1. verification of the current captive-carry survival function;
2. investigation of other methods for estimating the captive-carry survival function;
3. formulation of a strategy for using and retiring, or refurbishing AMRAAMs.

1.2.1 Operationally Assessing Captive-Carry Lifelength. When the USAF and USN field missiles, operational personnel assess the readiness of missiles via the BIT. As shown in section 1.1.3, the *true* captive-carry lifelength is the amount of time that a missile can be captive-carried and still remain functional. However, since only a portion of the missile can be tested (via BIT) while being flown, it is convenient to *operationally* define captive-carry lifelength as the cumulative length of time that a missile can be carried while each BIT testable subsystem remains functional[4]. Unless otherwise stated, this operational definition of captive-carry lifelength will be used for the remainder of the paper.

1.2.2 Current Captive-Carry Survival Model. Currently, AMRAAM engineers assume an exponential captive-carry survival function. The engineers model the captive-carry lifelength of an AMRAAM as an exponential random variable with parameter, $\lambda = 1/$captive-carry mean time between failure (MTBF). The exponential assumption was chosen based on experience with past missile analysis (e.g. relatively low probability (0.37) of remaining operational beyond the captive-carry MTBF) and ease of application. As a consequence of current data collection methods, complete verification of the exponential assumption has not been possible. Captive-carry lifelength data has been subject to partial *censoring*; that is, the captive-carry lifelength for a portion of missiles "cannot be fully observed" (Lagakos, 1979). Specifically, captive-carry lifelength data has

[4]As shown in section 1.1.3, AMRAAM has a degraded mode (DMA message: Failure - No Mission Impact); that is, the missile is still considered launch capable with certain component failure(s). For purposes of this analysis though, the missile has failed; unless fired during the sortie or the supply of replacement missiles is used up, the missile would be replaced for the next sortie and the component repaired.

been subject to both *interval and right censoring.* Interval censoring means that exact captive-carry lifelength of an AMRAAM that fails is not known, but an interval of time in which the missile fails is known. Right censoring occurs when a missile failure time has not yet been observed.

1.2.3 AMRAAM Captive-Carry Data. The USAF and USN use two main data sources to estimate the captive-carry survival function:

1. Product Reliability Acceptance Testing (PRAT) and
2. Operational Flights.

The USAF and USN use PRAT as a criteria for government acceptance or rejection of AMRAAM lots manufactured by the government's two production contractors: Hughes Missile Systems Company (HMSC) and Raytheon Company. For each contractor's production lot, the government randomly selects "a sample of representative missiles" for PRAT (NAWC, 1992). PRAT consists of a physical mock-up of captive-carry in which the sample missiles are placed in pairs on hooded shaker tables in temperature-adjustable rooms and subjected to changes in vibration and temperature which simulate a typical flight profile. Testing continues until the missiles fail or until a predetermined[5] number of test steps are completed. The government then calculates the MTBF of the sample using the exponential survival model (refer to section 2.2.1 for further details). The government uses the sample MTBF as an estimate of the production lot's MTBF.

PRAT data consists of interval and right censored data points. For each missile, test personnel perform BIT at predetermined points in the test process. When a BIT detects missile failure, the actual failure time occurs at a point in time between the previous BIT and this current BIT; missile failure times are thus interval censored. If BIT does not detect a failure for the duration of the test, the missile's failure time remains unobserved (right censored).

[5]Determined before test is conducted.

The USAF and USN also maintain an Operational Flight data base of the operational captive-carry hours for fielded AMRAAMs. Using the exponential survival model, the government estimates the MTBF of fielded missiles. Operational Flight Data also consists of interval and right censored data points. For each missile, ground personnel normally perform a BIT before and after a sortie and the aircrew may perform several BITs during the sortie. When a BIT detects missile failure, the actual failure time occurs at a point in time between the previous BIT and the current BIT. Thus, the missile failure times are interval censored unless the previous BIT (pass indication) is performed after a sortie and the current BIT detects the failure before the next sortie. As with the PRAT data, if BIT does not detect a failure during the time period the data is collected, then the missile failure times are right censored.

For the right censored observations, the current approach for both PRAT and Operational Flight data is to record the sum of the unobserved failure times as a single unobserved failure time (that is, a single right censored observation). However, for the uncensored observations, the current approach differs. For PRAT, the approach is to record an uncensored observation as the midpoint of the censoring interval (that is, record the time corresponding to the midpoint between the previous BIT (pass indication) and the current BIT (failure indication)). For Operational Flight Data, the approach is to record an uncensored observation using the time of first detection of the failure (that is, the right endpoint of an interval censored observation).

Although this decision at first glance seems naïve, it is correct if the exponential assumption of the captive-carry survival function is plausible. With the data structures as defined above, the occurrence of failures can be modeled as Poisson Process $\{N(t), t \geq 0\}$ where $N(t)$ is a random variable representing the number of missile failures by time t. The rate of this process is λ where λ is the calculated interarrival rate of a failure. As an example, suppose we observe (interval censored) captive-carry lifelengths of 300 hours, 250 hours, and 350 hours. Suppose further we observe 350, 250, and 400 captive-carry hours (right censored observations) without a failure. The total captive-

carry time is 1900 hours and the interarrival rate of failure, λ, is 0.00157 failures per hour (that is, we have observed three failures in 1900 hours). The number of failures in any time period K is $K\lambda$. One problem with this approach is that biasing is introduced into the estimate of the parameter λ by treating the interval censored observations as uncensored observations. However, since the interval of censoring is small in magnitude (e.g about 0.75 hours for PRAT) compared to the estimate of MTBF $(1/\lambda)$, the bias introduced is relatively small. Perhaps more importantly the validity of the exponential assumption should be formally checked, since the above analysis *requires the assumption that the missiles do not age* (constant failure rate).

1.3 Scope

1.3.1 Modeling the Captive-Carry Survival Function at the System versus Component Level.

BIT registers a system failure if one (hardware and/or software) component tested by BIT does not function; the missile does not have redundant components. Accordingly, the missile can be modeled as a system of n independent components in series. Defining the component level to be the unit level, BIT checks (at least partially) 17 of 20 units (reference section 1.1.3, Table 1). For instance, referring to Table 1, BIT checks all units of the guidance section except the unit Forward, Aft Fuselage and Miscellaneous Cables.

Let random variable T represent the captive-carry lifelength of an AMRAAM. Further, let the cumulative captive-carry failure function for component i be $F_i(t)$, $i = 1, 2, \ldots, 17$. The captive-carry survival function for component i (probability that component i survives for time greater than t), denoted by $S_i(t) = 1 - F_i(t)$, $i = 1, 2, \ldots, 17$. Having assumed the individual component lifetimes are independent, the captive-carry survival function for the missile, denoted $S(t)$, is the product of the component captive-carry survival functions; that is,

$$S(t) = \prod_{i=1}^{17} S_i(t).$$

The captive-carry lifelength for the missile equates to the minimum captive-carry lifelength for the components. No additional information will be gained at the system level by analyzing the captive-carry survival function for each component. Consequently, this paper limits the scope of analysis of the captive-carry survival function to the system level.

1.3.2 Modeling Approach. The following is an outline of the models used to estimate the captive-carry survival function:

- Poisson / Exponential Model
- Non-Parametric Models
- Other Applicable Parametric Models

These models are applied on two data sets. The first set contains PRAT of HMSC and Raytheon Company production lots 2 (sublot 3 only), 3, 4, 5, and 6 (sublot 1 only). The second set consists of operational flight data for USAF HMSC missiles based in Italy, Saudi Arabia, and Turkey. For the PRAT data, the failure times which are interval censored have been approximated by the midpoint of the interval. For the Operational Flight data, the failure times which are also interval censored have been approximated by the right endpoint of the interval (no information on the sortie lengths is available). Since the length of the largest censored interval is three orders of magnitude smaller than the current estimated MTBF for both data sets, the bias introduced into the estimation should be minimal.

The Poisson / Exponential model (with parameter $\lambda = 1/\text{MTBF}$) is duplicated in order to verify the current estimation of the captive-carry survival function. This model provides a baseline upon which the other models are compared. If the exponential assumption appears invalid, non-parametric analyses are used to determine if more appropriate parametric model(s) can be fit to the data sets. An alternative non-parametric model given by Lawson (1994) using Gibbs Sampling

(Casella and George, 1992) within a Bayesian framework will also be used to estimate the captive-carry survival function.

The Kaplan-Meier survival curve (Kaplan and Meier, 1958) and the bootstrapping of the Kaplan-Meier estimator (Efron, 1979 and Efron, 1981) provide a non-parametric estimate of the captive-carry survival function. A hazard plotting technique (Nelson, 1972) furnishes a non-parametric estimate of the cumulative captive-carry hazard function, denoted $H(t)$. The relationship between $H(t)$ and the captive-carry survival function, denoted $S(t)$, is $H(t) = -\log S(t)$. These estimates provide a point of verification for the exponential assumption. Additionally, the Mantel (1966) or log rank test and the Peto-Peto (1972) modification of Wilcoxon test along with Cox Proportional Hazards Model parameter tests (Kablfleish and Prentice, 1980) are used to determine whether or not the samples based on sublot or lot and flying region have arisen from identical or separate survival functions.

1.4 Summary

The thesis considers the problem of estimating the survival function of an item (probabilityy that the item functions for a time greater than a given time t) from sampling data subject to partial right censoring (a portion of the items in the sampling data have not yet been observed to fail). The thesis is written in a *case study* format. Specifically, the case of estimating the captive-carry survival function of the AIM-120A Advanced Medium Range Air-to-Air Missile (AMRAAM) is considered. This chapter introduced the problem of estimating the captive-carry survival function for the AMRAAM, providing an overview of the problem and complications introduced by censored sampling data. The remainder of the thesis is divided into four chapters. Chapter 2 (Literature Review) provides a detailed description of the statistical models and censoring issues introduced in Chapter 1. Chapter 3 (Methodology) details the specific methodology used in estimating the captive-carry survival function, providing a description of the sample data and the algorithms for

the statistical models. Chapter 4 (Findings) documents the results of the statistical analysis of the captive-carry survival function. Chapter 5 (Conclusion) outlines the conclusions that may be drawn from the statistical analysis and provides a taxonomy for estimating the survival function of an item from sampling data subject to partial right censoring.

II. Literature Review

2.1 Introduction

The literature review provides a summary of statistical research in the area of distribution estimation from sampling data subject to right censoring. The author searched the Citation Index for Statistics (CIS)[1] to formulate the summary. This review includes statistical research deemed relevant to the problem of estimating the captive-carry survival function. This chapter is divided into two sections. Section 2.2 (Discussion of Literature) provides an overview of literature on right censoring and the parametric and non-parametric models for estimating survival functions when the sampling data is subject to right censoring. Section 2.3 (Summary) provides a brief summary of the material in section 2.2. (The reader may find it of benefit to read this summary before proceeding to the section 2.2.)

2.2 Discussion of Literature

Statistical models for estimating distributions are usually categorized as either parametric or non-parametric. Miller observes that

> *Most of the [early] statistical research for engineering applications was concentrated on parametric models. Within the past two decades there has been an increase in the number of clinical trials in medical research, and this has shifted the statistical focus to non-parametric approaches. (Miller, 1981, page 2)*

As discussed in Chapter 1, the military services currently model the captive-carry survival function using a parametric model (Poisson / Exponential Model); however, the assumption of the exponential distribution of failure times that underlies the model has not been verified. Non-parametric models such as the Kaplan-Meier survival curve (Kaplan-Meier, 1958) and Nelson's hazard plotting technique (Nelson, 1972) provide a graphical means to verify the reasonableness of exponential assumption. These approaches can also provide consistent estimates in there own right.

[1] The CIS provides indexing coverage for the field of statistics.

2.2.1 Right Censoring. Recall that random variable T represents the captive-carry lifelength associated the captive-carry survival function $S(t)$. An uncensored sample of size n consists of $T_1, T_2, \ldots, T_n$ which are independent, identically distributed (IID) observations from distribution S.

When the sample is subject to right censoring, T_i is only observable for the failed missiles. In order to model data subject to right censoring, it is necessary to consider the nature of the censoring mechanism. Specifically, let random variable C represent the censoring time and let random variable Y represent the observed portion of the captive-carry lifelength. Assuming the T and C are independent (referred to as *independent censoring*), a right censoring sample of size n consists of observations $Y_1, Y_2, \ldots, Y_n$ where $Y_i = \min(T_i, C_i)$. Equivalently, let d represent an indicator variable such that

$$d_i = \begin{cases} 1 & \text{if the observation } Y_i \text{ is not censored and} \\ 0 & \text{if the observation } Y_i \text{ is right censored,} \end{cases}$$

a right censoring sample of size n consists of $(Y_1, d_1), (Y_2, d_2), \ldots, (Y_n, d_1)$ where

$$d_i = I(T_i \leq C_i) = \begin{cases} 1 & \text{if } T_i \leq C_i \text{ and} \\ 0 & \text{if } T_i > C_i. \end{cases}$$

Lagakos (1979) provides a "full probabilistic description" of the observable pair (Y,d) given p, F_D, and F_C where

- $p = P(d = 1)$ is the probability that an observation is uncensored;
- $F_D(t) = P(Y \leq t \mid d = 1)$ is the conditional cumulative time-to-failure distribution of the observation given that the observation is uncensored; and
- $F_C(t) = P(Y \leq t \mid d = 0)$ is the conditional distribution of observation given the observation is right censored.

Suppose further that events D_y and C_y respectively denote the occurrence of an uncensored and a censored observation in a neighborhood of time y. Then

$$P(D_y) = P(Y \in N_y, d = 1) = pdF_D(y) \text{ and } P(C_y) = P(Y \in N_y, d = 0) = (1-p)dF_C(y)$$

where $N_y = (y - dy, y)$ and $dF_j(y) = F_j^{'}(y)dy$ for $j = D, C$. Given a set of independent realizations of (Y, d), it follows that the subsample of failure times is a set of observations from F_D and the subsample of right censored times form a set of observations from F_C. What is important to realize is the subsample of failure times is not a sample from the $F(t)$. A set of right censored data only provides information about p,F_D, and F_C. The general problem that right censoring introduces is relating this information to $F(t)$ and $S(t)$.

2.2.2 Parametric Models. A typical treatment of the Poisson Process can be found in Ross, (1983) and (1993). The Poisson Process is a counting process $\{N(t), t \geq 0\}$ having rate λ, $\lambda > 0$, where $N(t)$ represents the total number of events that have occurred up to time t. In our case, $N(t)$ represents the number of missiles that have failed up to time t.

The interarrival times of the Poisson Process are independent identically distributed exponential random variables having mean $1/\lambda$. An interarrival time is defined as the elapsed time between events or, in our case, the elapsed time between AMRAAM missile failures. Hence, when utilizing the Poisson Process model, the captive-carry survival function for the AMRAAM is assumed to be exponential. As discussed in section 1.2.3, the military services record the sum of the unobserved failure times as a single unobserved failure time (i.e., a single right censored observation). The memoryless property of exponential provides for this summation; that is, the distribution of the remaining lifetime is independent of the amount of time that the object has already survived.

The method of maximum likelihood provides an estimator for the parameter λ, of the exponential model. Lagakos (1979) shows that the likelihood function for the right (independent)

censoring model given in section 2.2.1. is proportional to

$$\{\prod_j dF(t_j)\}\{\prod_k S(r_k)\}\{\prod_j(1 - H(t_j)\}\{\prod_k dH(r_k)\}. \tag{1}$$

where t_j are observed failure times and r_k are right censored observations. Since the censoring function $H(t)$ is independent from the survival function $S(t)$, the last two products in equation 1, $\{\prod_j(1 - H(t_j)\}$ and $\{\prod_k dH(r_k)\}$, do not involve the unknown parameter λ. Hence, these two products can be treated like constraints when maximizing the likelihood function.

Miller (1981) provides the following derivation of the maximum likelihood estimator for the parameter λ, $\max_\lambda L(\lambda)$. Finding $\max_\lambda L(\lambda)$ is equivalent to finding the solution $\hat{\lambda}$ to the likelihood equation

$$0 = \frac{\partial}{\partial \lambda} \log L(\lambda) = \sum_j \frac{\partial}{\partial \lambda} \log dF(t_j) + \sum_k \frac{\partial}{\partial \lambda} \log S(r_k). \tag{2}$$

For the exponential model, $dF(t_j) = \lambda \exp^{(-\lambda t_j)}$ and $S(r_k) = \exp^{(-\lambda r_k)}$, the equation 2 reduces to

$$0 = \frac{\partial}{\partial \lambda} \log L(\lambda) = \sum_j \frac{\partial}{\partial \lambda}(\lambda - \lambda t_j) + \sum_k \frac{\partial}{\partial \lambda}(-\lambda r_k). \tag{3}$$

Solving for λ in equation 3 yields

$$\hat{\lambda} = \frac{j}{\sum_j t_j + \sum_k r_k}.$$

Thus the maximum likelihood estimator $\hat{\lambda}$ equates to the number of observed AMRAAM failures divided by the summation of all observed and right censored captive-carry lifelengths.

2.2.3 Non-parametric Models. The Kaplan-Meier Survival Curve is a non-parametric estimate of S(t) (Kaplan and Meier, 1958). Suppose we observe a sample of n failure times (e.g., AMRAAM captive-carry lifelengths). The Kaplan-Meier Survival Curve is formed by giving mass $1/n$ to each observed value in the sample. As justification for curve formulation, Kaplan and Meier reaffirm the statistical finding that the maximum likelihood, non-parametric estimator of the

survival function is the empirical distribution function. By the Clivenko-Cantelli Lemma (Chung, 1974), the estimator is also consistent since the empirical distribution function F_n converges in distribution to F where $F(t) = 1 - S(t)$ for all t.

When a portion of a sample is right censored (m uncensored observations and c right censored observations), the Kaplan-Meier Survival Curve is formed by

1. dividing the m uncensored observations into intervals $(0, u_1), (u_1, u_2), \ldots, (u_2, u_m)$ such that each failure occurs at $u_i, i = 1, 2, \ldots, m$;
2. estimating p_i, the proportion of items alive just after u_{i-1} that survive beyond u_i, for each interval; and
3. estimating $S(t)$ as the product of the estimated p_i for all intervals prior to t.

Miller (1983) warns the scientific community against blindly employing the Kaplan-Meier Survival Curve without considering the actual sample data at hand. As stated by Miller, the Kaplan-Meier Survival Curve is "attractive because it is easy to compute and understand". Because the technique does not assume an underlying probability structure, "no assumptions are required other than the basic one of independence between the survival and censoring variables". However, as Miller points out, parametric modeling, if applicable, will increase "the precision in the estimation of probabilities". On the other hand, if the assumed underlying parametric model is incorrect, the parametric estimate will not be consistent.

Efron (1979) introduces the non-parametric bootstrapping technique. The bootstrap, a data-based simulation method for statistical inference, produces inferences such as standard deviations, confidence intervals, biases, and so forth. Figure 2 (Efron and Tibshirani, 1993) shows a schematic diagram of the bootstrap method as it applies to the one sample problem.

On the left is the real world, where an unknown distribution F (in our case, the cumulative captive-carry failure function) gives observed data, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, (the sample captive-carry

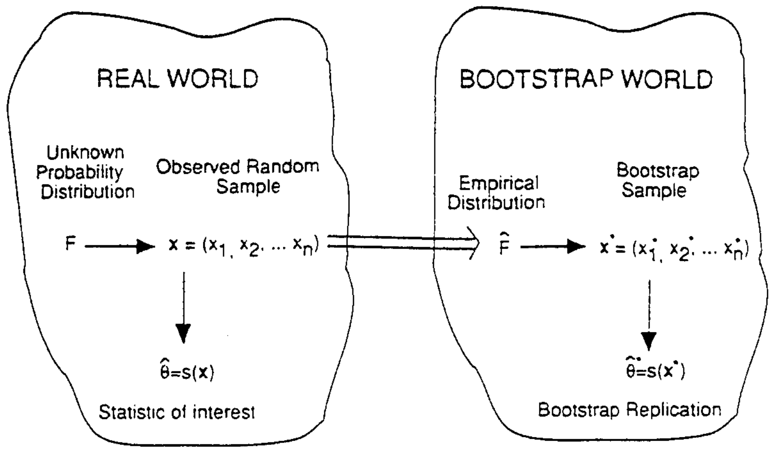


Figure 2. Bootstrap Method for the One Sample Problem

lifelengths) by random sampling. The statistic, $\hat{\theta} = s(\boldsymbol{x})$, estimates the population parameter of interest θ. On the right side of the diagram is the bootstrap world. Here the empirical distribution, $\hat{F}$, (which gives mass $1/n$ to each observed value) generates bootstrap samples $\boldsymbol{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$ from which the statistic $\hat{\theta}^* = s(\boldsymbol{x}^*)$ is calculated.

A bootstrap sample $\boldsymbol{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$ is obtained by randomly sampling n times, with replacement, from the observed data, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$. For instance with $n = 5$, one possible bootstrap sample is $\boldsymbol{x}^* = (x_5^*, x_2^*, x_3^*, x_2^*, x_4^*)$. The bootstrap method generates N independent bootstrap samples $\boldsymbol{x}^{*1}, \boldsymbol{x}^{*2}, \ldots, \boldsymbol{x}^{*N}$, each of size n. Corresponding to each bootstrap sample is a bootstrap replication of the statistic of interest, $s(\boldsymbol{x}^{*b})$. In general (assuming an empirical distribution of $\hat{F}$), the value $\hat{\theta}$ is then calculated by

$$\hat{\theta}^* = \frac{\sum_{b=1}^{N} s(\boldsymbol{x}^{*b})}{N}$$

Efron and Tibshirani highlight that the bootstrap method has two primary advantages:

1. the method is non-parametric, relieving the user from having to make parametric assumptions about the form of the underlying distribution F and

2. the method provides estimates of statistics of interest for which no closed form expressions exist (when no formula exists such that $\theta = s(\boldsymbol{x})$).

Efron (1981) extends the use of the bootstrap technique to right censored data. The bootstrap method for censored data parallels the general bootstrap method as outlined above with the Kaplan-Meier product limit estimator (Kaplan-Meier, 1958) as the statistic of interest θ.

Nelson (1972) proposes a simple graphical procedure for estimating of the cumulative hazard function , denoted $H(t)$. The procedure is non-parametric and, by virtue of the basic relationship $H(t) = -\log S(t)$, also provides a non-parametric estimate of the survival function. Nelson's procedure is accomplished as follows for a sample of size n:

1. Uncensored and right censored observations are grouped together and ordered with respect to magnitude (short time to largest time) and assigned reverse rank numbers, i.e., the first observation is assigned the rank n, the next is assigned the rank $n-1$. The process is continued until the last observation is assigned rank 1. The rank represents the number of survivors immediately prior to censoring of a corresponding observation.

2. Next, an estimate of the hazard rate, $h(t)$, in percent is calculated for all ordered observations $(y^i, i = 1, 2, \ldots, n)$ as

$$\hat{h}(y^i) = 100(\frac{1}{m_i})$$

   where m_i is the reverse rank corresponding to i.

3. An estimate of $H(t)$ is calculated for each y^i as

$$\hat{H}(y^i) = \sum_{j=1}^{i} \hat{h}(y^j)$$

4. A plot of $\hat{H}(t)$ versus t is made for the *uncensored* observations. If the correct scale is chosen for $\hat{H}(t)$ and t, then plotted points will lie on a straight line.

Nelson develops the plotting technique for the following distributions: exponential, weibull, extreme value, normal, and lognormal. Distribution parameters may be estimated directly from the plots. As an example, for the exponential distribution, $S(t) = \exp^{(-\lambda t)}$ and $H(t) = -\log S(t) = \lambda t$. $H(t)$ is a linear function of t. On the real coordinate axes, a plot of $\hat{H}(t)$ versus t for the exponential distribution appears as a straight line with the slope of the line corresponding to $\hat{\lambda}$.

Nelson assumes independent censoring. He presents a theoretical basis for the "reasonableness of hazard plotting positions" by showing that the procedure is "based on the properties of order statistics of Type II progressively censored samples". However, Nelson does not derive the result for the independent censoring case and states only that he expects "a similar result holds for randomly [independent] censored samples."

Another useful technique is Gibbs Sampling, defined by Casella and George (1992) as "a technique for generating random variables for a distribution indirectly without having to calculate the density". Specifically, let $f_{Y_1,Y_2,\ldots,Y_n}$ be the joint distribution of random variables $Y_1, Y_2, \ldots, Y_n$. If we know the conditional distributions $f_{Y_i|Y_j,j\neq i}$, $i = 1, 2, \ldots, n$, Casella and George show that observations can be generated from the conditional distributions $f_{Y_i|Y_j,j\neq i}$, $i = 1, 2, \ldots, n$ such that these observations come from a distribution approximately equal to the joint distribution $f_{Y_1,Y_2,\ldots,Y_n}$. To generate observations from $f_{Y_1,Y_2,\ldots,Y_p}$, the algorithm proceeds as follows. First fix arbitrary starting values $Y_1^{(0)}, Y_2^{(0)}, \ldots, Y_n^{(0)}$ and then update these values. Draw $Y_1^{(1)}$ from $f_{Y_1|Y_j,\,j\neq 1}(\cdot, Y_2^{(0)}, \ldots, Y_n^{(0)})$. Next, draw $Y_2^{(1)}$ from $f_{Y_2|Y_j,\,j\neq 2}(Y_1^{(1)}, \cdot, Y_3^{(0)}, \ldots, Y_n^{(0)})$. Continue until drawing $Y_n^{(1)}$ from $f_{Y_n|Y_j,\,j\neq n}(Y_1^{(1)}, \ldots, Y_{n-1}^{(1)}, \cdot)$. One iteration of the scheme is completed after visiting each variable. After K iterations, the random variables $(Y_1^{(K)}, \ldots, Y_n^{(K)})$ are generated. The sequence $(Y_1^{(j)}, Y_2^{(j)} \ldots, Y_n^{(j)})$, $j = 1, 2, \ldots, n$ is a Markov chain and $f_{Y_1,Y_2,\ldots,Y_n}$ is a stationary distribution of the chain.

Lawson (1994) uses a Gibbs Sampling algorithm to provide a non-parametric Bayesian estimate of the distribution of m independent component lifelengths in a coherent system G. Lawson's extension enables the construction of an algorithm for estimating the captive-carry survival function when the sampling data is subject to right censoring. The algorithm uses mixtures of Dirichlet priors (Ferguson, 1973 and 1974). In our case, consider the parametric family H_θ, $\theta \in \Theta$, and put a mixture of Dirichlets as the prior on F (where $F(t) = 1 - S(t)$). That is

$$F \sim \int \mathcal{D}_{\alpha_\theta}\, \nu(d\theta)$$

where for each $\theta \in \Theta$, $\alpha_\theta = \alpha_\theta(\mathcal{R})H_\theta$, $0 < \alpha_\theta(\mathcal{R}) < \infty$, and ν is a probability measure on Θ. Here, $\alpha_\theta(\mathcal{R})$ represents the degree of concentration of $\mathcal{D}_\alpha$ around its center H. The posterior of F (i.e., the conditional distribution of F given $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is

$$\int \mathcal{D}_{\alpha_\theta + \sum_{i=1}^{n} \delta_{x_i}}^{\nu_x^{(d\theta)}}$$

where δ is the Dirac delta function and ν_x is the posterior distribution of θ given $\boldsymbol{x}$.

The algorithm proceeds as follows. Let $u_j, j = 1, 2, \ldots, m$ be the uncensored observations in a sample and $r_i, i = 1, 2, \ldots, c$ be the right censored observations in the sample if we could observe them. Fix arbitrary starting values $z_1^{(0)}, z_2^{(0)}, \ldots, z_c^{(0)}$ such that $z_i^{(0)} > r_i$. Generate $z_1^{(1)} \sim \mathcal{L}_{data}(z_1 \mid u_1, u_2, \ldots, u_c, z_2^{(0)}, z_3^{(0)}, \ldots, z_c^{(0)})$. Next generate $z_2^{(1)} \sim \mathcal{L}_{data}(z_2 \mid u_1, u_2, \ldots, u_c, z_1^{(1)}, z_3^{(0)}, \ldots, z_c^{(0)})$ and continue until $z_c^{(1)} \sim \mathcal{L}_{data}(z_c \mid u_1, u_2, \ldots, u_c, z_1^{(1)}, z_2^{(1)}, \ldots, z_{c-1}^{(1)})$. One iteration of the procedure is completed. Repeat the procedure a large number of times and use realizations of the chain to estimate $\mathcal{L}_{data}(z_1, z_2, \ldots, z_c)$.

Lawson's non-parametric Bayesian procedure has two advantages. First, the procedure guards against the problems associated with using an incorrectly specified parametric model by using a mixture of Dirichlet priors. Second, the procedure avoids the loss of efficiency due to ignoring

partial information about a parametric model, since the the prior distributions concentrate their mass around the hypothesized parametric family.

The Mantel (1966) test or *log rank* test and Peto-Peto modification (1972) to the Wilcoxon test furnish methods to determine whether two or more samples could have arisen from identical survival functions. The tests are fairly intuitive since the test statistic is the difference between the observed number of failures in each sample and the expected (in an informal sense) number of failures under the null hypothesis that the samples are from the same survival function.

The log-rank test proceeds as follows (Kalbfleish and Prentice, 1980). To test the equality of the survivor functions $S_1(t), S_2(t), \ldots, S_r(t)$ on the basis of samples from each of r populations, let $t_1, t_2, \ldots, t_k$ denote the failure times formed by pooling the r samples. Suppose d_j failures occur at t_j and that n_j items are at "risk" just prior to $t_j, j = 1, 2, \ldots, k$. Let d_{ij} and n_{ij} be the corresponding numbers in sample i, $i = 1, 2, \ldots, r$. Conditional on the failure and censoring experience up to time t_j the probability mass function of $d_{1j}, d_{2j}, \ldots, d_{rj}$ is simply the product of binomial distributions

$$\prod_{i=1}^{r} \begin{pmatrix} n_{ij} \\ d_{ij} \end{pmatrix} \lambda_j^{d_j} (1-\lambda_j)^{n_j - d_j}$$

where λ_j is the conditional failure time at t_j, which is common for each of the r samples under the null hypothesis. The conditional probability mass function for $d_{1j}, d_{2j}, \ldots, d_{rj}$ given d_j is then the hypergeometic distribution

$$\frac{\prod_{i=1}^{r} \begin{pmatrix} n_{ij} \\ d_{ij} \end{pmatrix}}{\begin{pmatrix} n_j \\ d_j \end{pmatrix}}. \tag{4}$$

The mean and variance of d_{ij} from equation 4 are, respectively,

$$w_{ij} = n_{ij} d_j n_{ij}^{-1}$$

and

$$(V_j)_{ii} = n_{ij}(n_j - n_{ij})d_j(n_j - d_j)n_j^{-2}(n_j - 1)^{-1}.$$

The covariance of d_{ij} and d_{lj} is

$$(V_j)_{il} = -n_{ij}n_{lj}d_j(n_j - d_j)n_j^{-2}(n_j - 1)^{-1}.$$

Thus the statistic $\boldsymbol{v}_j^T = (d_{1j} - w_{1j}, d_{2j} - w_{2j}, \ldots, d_{rj} - w_{rj})$ has (conditional) mean zero and variance matrix, V_j. A summation over failure times gives the log rank statistic

$$\boldsymbol{v} = \sum_{j=1}^{k} \boldsymbol{v}_j \tag{5}$$

which is the vector of the observed number of failures in each sample minus the corresponding vector of the *expected* failures. However, the vector of expected failures is not an overall expected number of failures but is rather the sum over failure times of the conditional expected number of failures in each sample given the total number of failures at that time.

The asymptotic results of partial likelihood theory (Cox, 1975) can be used to show that $\boldsymbol{v}$ is asymptotically normal with estimated covariance matrix V, where

$$V = \sum_{j=1}^{k} V_j.$$

An appropriate test of equality of the r survival functions can be based on an asymptotic distribution χ^2_{r-1} distribution for

$$\boldsymbol{v}V^{-1}\boldsymbol{v}. \tag{6}$$

The Peto-Peto modification to the Wilcoxon test weights the statistic $\boldsymbol{v}$ in equation 5 at each failure time t_j by the Kaplan-Meier product limit estimator (denoted $S_{km}(t_j)$); that is

$$\boldsymbol{v} = \sum_{j=1}^{k} S_{km}(t_j)\boldsymbol{v}_j. \tag{7}$$

The test statistic is also given by equation 6.

The Cox Proportional Hazards Model (Cox, 1972) enables the regression of predictor variables against response variables when the sampling data is subject to independent right censoring. The proportional hazards model is specified by the non-parametric hazard relationship

$$\lambda(t;\boldsymbol{z}) = \lambda_0(t)\exp^{(\boldsymbol{z}\boldsymbol{\beta})} \tag{8}$$

where $\boldsymbol{z}$ is row vector of s measured variables or covariates, $\boldsymbol{\beta}$ is a column vector of s regression parameters, T is the associated failure time, and $\lambda_0(t)$ is an arbitrary and unspecified baseline hazard function.

Kablfeish and Prentice (1980) detail a number of approaches to estimating the parameters $\boldsymbol{\beta}$ in equation 8. The main approaches used are the method of marginal likelihood and partial likelihood. Kablfeisch and Prentice state,

> *Historically, the method of partial likelihood was the first applied to the [proportional hazards] model and in many ways is the most general of the methods outlined here. The method of marginal likelihood is discussed on account of its close relationship to the [log] rank tests ... and also because of the directness of the marginal likelihood solution.*

Kablfeisch and Prentice derive the following likelihood expression for $\boldsymbol{\beta}$ using the method of marginal likelihood. Suppose that m items are uncensored and c items are right censored. Suppose that of the m items, items $i1, i2, \ldots, id_i$ are observed to fail at $t_{(i)}, i = 1, 2, \ldots, k$ where $t_{(1)} < t_{(2)} <, \ldots, < t_{(k)}$ and $\sum d_i = m$. If the number of items failing, d_i, at each failure point is small compared to the number of items in risk set before $t_{(i)}$, denoted $R(t_{(i)})$, then the marginal

likelihood of $\boldsymbol{\beta}$ will be approximated by

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{k} \frac{\exp^{(\boldsymbol{s}_i \boldsymbol{\beta})}}{[\sum_{l \in R(t_{(i)})} \exp^{(\boldsymbol{z}_l \boldsymbol{\beta})}]^{d_i}} \tag{9}$$

where the risk set $R(t_{(i)})$ consists of items *alive* prior to $t_{(i)}$ and $\boldsymbol{s}_i = \sum \boldsymbol{z}_{ij}$ is the sum of the covariates of items observed to have failed by $t_{(i)}$. The maximum likelihood estimate $\hat{\boldsymbol{\beta}}$, from equation 9 can be obtained as a solution the system of equations

$$U_j(\boldsymbol{\beta}) = \frac{\partial \log L}{\partial \beta_j} = \sum_{i=1}^{k} [s_{ji} - d_i A_{ji}(\boldsymbol{\beta})] = 0 \qquad (j = 1, 2, \ldots, s) \tag{10}$$

where s_{ji} is the jth element in $\boldsymbol{s}_i$ and

$$A_{ji}(\boldsymbol{\beta}) = \frac{\sum_{l \in R(t_{(i)})} z_{jl} \exp^{(\boldsymbol{z}_l \boldsymbol{\beta})}}{\sum_{l \in R(t_{(i)})} \exp^{(\boldsymbol{z}_l \boldsymbol{\beta})}}$$

In addition,

$$I_{hj}(\boldsymbol{\beta}) = -\frac{\partial^2 \log L}{\partial \beta_h \beta_j} = \sum_{i=1}^{k} d_i C_{hji} \tag{11}$$

where

$$C_{hji} = \frac{\sum_{l \in R(t_{(i)})} z_{hl} z_{jl} \exp^{(\boldsymbol{z}_l \boldsymbol{\beta})}}{\sum_{l \in R(t_{(i)})} \exp^{(\boldsymbol{z}_l \boldsymbol{\beta})}} - A_{hi}(\boldsymbol{\beta}) A_{ji}(\boldsymbol{\beta}) \qquad (h, j = 1, 2, \ldots, s)$$

The value $\hat{\boldsymbol{\beta}}$ that maximizes equation 9 can be obtained by a Newton-Raphson iteration utilizing equation 10 and equation 11. Only mild conditions on the covariates and censoring are required to ensure the asymptotic normality of $\hat{\boldsymbol{\beta}}$. In the absence of ties in the failure times, this asymptotic normal distribution has mean $\boldsymbol{\beta}$ and estimated covariance matrix $I(\hat{\boldsymbol{\beta}})^{-1} = [I_{hj}(\hat{\boldsymbol{\beta}})]^{-1}$. The same asymptotic results hold with tied failure times except that there is some asymptotic bias in both the estimation of $\hat{\boldsymbol{\beta}}$ and the covariance matrix owing to the approximation used in equation 9.

Kalbfleisch and Prentice 1980 detail three procedures for testing $\beta = 0$ for a single covariate z. The first test is based on the asymptotic normal distribution of β; that is, under the assumption that $\beta = 0$, $\hat{\beta}\sqrt{I(\hat{\beta})}$ is an observation from a $N(0,1)$ distribution. The second test of $\beta = 0$ is based on the likelihood ratio

$$R(\theta) = \frac{L(\theta)}{L(\hat{\theta})}$$

and its asymptotic distribution. If the regularity conditions of maximum likelihood theory hold, then under the hypothesis $\beta = 0$, the asymptotic distribution of $-2\log R(0)$ is χ^2 with one degree of freedom. A third procedure for testing $\beta = 0$ is by the asymptotic distribution of the score statistic $U(\theta)$ which equals $\sum_{i=1}^{n} U_i(\theta)$ where

$$U_i(\theta) = \frac{\partial}{\partial\theta}\log L_i(\theta)$$

$i = 1, 2, \ldots, n$. Specifically, under the hypothesis $\beta = 0$, U(0) is asymptotically normal with mean 0 and estimated variance $I(0)^{-1}$. Consequently, $U(0)^2 I(0)^{-1}$ has an asymptotic χ^2 distribution with one degree of freedom. As a final note, to check whether the large sample likelihood assumptions are reasonably accurate, Kabfleisch and Prentice recommend comparing a plot of the log relative likelihood versus β

$$R(\beta) = \log L(\beta) - \log L(\hat{\beta})$$

to a plot of the normal likelihood versus β

$$-\frac{1}{2}I(\hat{\beta})(\hat{\beta} - \beta)^2.$$

If the plots show close agreement then large sample likelihood assumptions should be reasonably accurate.

2.3 Summary

This chapter detailed relevant statistical research in the area of distribution estimation from sampling data subject to right censoring. Statistical models for estimating distributions are usually categorized as either parametric or non-parametric. Of all parametric models, the Poisson process is the oldest and most established model. The Poisson process is a counting process $\{N(t), t \geq 0\}$ having rate λ, $\lambda > 0$, where $N(t)$ represents the total number of events that have occurred up to time t. The interarrival times of the Poisson Process are independent identically distributed exponential random variables having mean $1/\lambda$. The maximum likelihood estimator of λ in the presence of right independent censoring is the number of observed failures divided by the summation of all observed failures and right censored observations.

The non-parametric models include the Kaplan-Meier's Product Limit Survival Curve (Kaplan and Meier, 1958); Efron's bootstrap estimate of the Kaplan-Meier estimator (Efron, 1981); Nelson's (1972) cumulative hazard function plotting technique; Lawson's Bayesian non-parametric estimator via the Gibbs Sampling algorithm (Lawson, 1994); and Cox Proportional Hazards Model (Cox, 1972). The Kaplan-Meier Survival Curve estimates the cumulative survival function, $S(t)$, by placing mass at the observed failure times and weighting these times based on the number of observations (censored and uncensored) at "risk" prior to a given failure time. The bootstrap is a data-based simulation technique for statistical inference that uses data resampling to produce inferences such as standard deviations. In the case of independent right censored data, the Kaplan-Meier product limit estimator is generated N times by resampling with replacement from the original data set.

Nelson's cumulative hazard plotting technique provides a non-parametric estimate of the cumulative hazard function, $H(t)$. By virtue of the basic relationship $H(t) = -\log S(t)$, the technique also provides an estimate of the survival function. An estimate of $H(t)$ is made in a manner similiar to the Kaplan-Meier product limit estimator. A plot of $\hat{H}(t)$ versus t is made for the uncensored

observations. If the correct scale is chosen for $\hat{H}(t)$ and t, then the underlying distribution can be determined. Lawson's Bayesian non-parametric estimator uses a Gibbs Sampling algorithm to generate "true" failure times from the right censored observations. The method uses a mixture of Dirichlet priors in conjunction with repeated draws from conditional distributions to generate these failure times. The Cox Proportional Hazards Model enables regression of predictor variables against response variables when the sampling data is subject to independent right censoring. The model can be used to assess whether two or more samples could have arisen from identical survival functions. The log rank (Mantel, 1966) and Peto-Peto modification to the Wilcoxon test also provide a means to test this assessment.

III. Methodology

3.1 Introduction

This chapter provides a description of the methodology used for this thesis effort. The chapter contains five sections. Section 3.2 (Notation) provides a list of the notation used in the remainder of the thesis. Section 3.3 (Sampling Data) details the samples used in estimating the captive-carry survival function. Section 3.4 (Sampling Data Constraints) explains the constraints placed on the analysis by the sampling data. Section 3.5 (Approach) reviews the objectives of the analysis, explains how the models are used to obtain the objectives; and details the implementation of the models. Section 3.6 (Summary) provides a summary of the chapter.

3.2 Notation

The following notation is used for the remainder of the thesis

- $F(t)$ denotes the cumulative captive-carry failure function;
- $S(t)$, $(1 - F(t))$ denotes the captive-carry survival function;
- n denotes a sample of size n of partially censored observations corresponding to a subsample of m uncensored observations (observed failures) and a subsample of c right censored observations;
- y_i denotes an observation from a sample;
- $h(t)$ denote the captive-carry hazard rate; and
- $H(t)$ denotes the cumulative captive-carry hazard function.

3.3 Sampling Data

As set forth in section 1.2.3, the military services use Product Reliability Acceptance Testing (PRAT) and Operational Flight data to estimate the captive-carry survival function. Table 2 lists

PRAT Samples

Sample	Sample Size (HMSC)	Sample Size (Raytheon)
Production Lot 2, Sublot 3	8	10
Production Lot 3, Sublot 1	10	10
Production Lot 3, Sublot 2	10	10
Production Lot 4, Sublot 1	12	12
Production Lot 4, Sublot 2	11[1]	12
Production Lot 5, Sublot 1	11	12
Production Lot 5, Sublot 2	10	11
Production Lot 6, Sublot 1	12	12
Total	84[2]	89

1. In the *BIT* analysis, the sublot sample size is 12.
2. In the *BIT* analysis, the total sample size is 85.

Operational Flight Samples (HMSC Missiles)

Italy		Saudi Arabia		Turkey	
Production Lot	Sample Size	Production Lot	Sample Size	Production Lot	Sample Size
Lot 3	29	Lot 3	75	Lot 2	34
		Lot 5	40	Lot 3	45
				Lot 6	12

Table 2. PRAT and Operational Flight Sampling Data

the PRAT and Operational Flight sampling data that is analyzed in this thesis effort. The PRAT sampling data consists of production sublot samples from the missile manufacturers, Hughes Missile System Company (HMSC) and Raytheon Company. For the purposes of PRAT, the services divide each production lot into two or more sublots. This division enables testing of potential manufacturing and performance differences across production lots (AMRAAM, 1993). The Operational Flight sampling data consists of *HSMC* missiles flown in sorties originating in Italy, Saudi Arabia, and Turkey. The field units that reported the Operational Flight sampling data aggregated the data at the production lot level (the reported data was not grouped by sublot).

As discussed in section 1.2.3, PRAT consists of a captive-carry simulation in which the sample missiles are placed in pairs on hooded shaker tables in temperature-adjustable rooms and subjected

Step	Step Time (min)	Cumulative Time (min)	Temperature (F°)	Vibration Multiplier
1	16.8	(16.8)	+97	0.09
2	15.0	(31.8)	-57	0
3	23.6	(55.4)	-57	0.09
4	16.0	(71.4)	-40	0.09
5	14.0	(85.4)	-31	0.09
6	0.4	(85.8)	-31	0.38
7	10.8	(96.6)	-8	0.09
8	1.2	(97.8)	-8	0.24
9	0.4	(98.2)	-8	0.38
10	45.0	(143.2)	+5	0.24
11	5.8	(149.0)	+5	0.24
12	15.0	(164.0)	+121	0
13	10.2	(174.2)	+121	0.24
14	0.2	(174.4)	+121	0.85
15	2.4	(176.8)	+121	0.38
16	16.8	(193.6)	+59	0.09
17	9.0	(202.6)	+50	0.38
18	1.8	(204.4)	+50	0.56
19	0.8	(205.2)	+50	0.50
20	34.0	(239.2)	+38	0.09
21	11.2	(250.4)	+43	0.24
22	6.2	(256.6)	+97	0.09
23	16.8	(273.4)	+59	0.09
24	30.0	(303.4)	+121	0
25	10.0	(313.4)	+70	0

The initial temperature setting for a cycle is offset repetitively as follows: $0F^{\circ}$, -$6F^{\circ}$, -$25F^{\circ}$, -$14F^{\circ}$, $5F^{\circ}$, $19F^{\circ}$, $31F^{\circ}$, $12F^{\circ}$.

Table 3. PRAT Simulated Flight Profile

to changes in vibration which simulate a typical flight profile. Specifically, the flight profile or test cycle consists of 25 different steps totaling 313.4 minutes (5.22 hours) including 243.4 minutes of vibration/thermal test time and 70.0 minutes of thermal (only) conditioning test time. Table 3.3 details the vibrational and temperature settings for the simulated flight profile.

The test personnel cycle the flight profile to simulate repetitive captive-carry flights (AMRAAM, 1993). The specific number of test cycles each missile undergoes is set arbitrarily based on a pretest specification of total test hours for the sublot sample. For instance, if PRAT consists of at least 1000 hours of total test time on three missiles, then missile 1 might be tested for 77 cycles (401.94 hours); missile 2, 58 cycles (302.76 hours); and missile 3, 57 cycles (297.54 hours).

Additionally, as a direct result of testing missiles in pairs, it is not uncommon to observe missiles with the same number of test hours.

The missiles used in PRAT are the same configuration as operational missiles with the following exceptions:

- the armament section is replaced with a telemetry section. The telemetry section is ballasted to armament section weight and balance.
- the propulsion section is inert.

Since the armament and propulsion sections are not tested during BIT (reference section 1.1.3, Table 1), this configuration change does not impact status information provided by BIT. During a cycle, test personnel conduct a BIT five times during a cycle at the following cumulative cycle times (in minutes): 5, 36.4, 140, 210, and 300 (NAWC, 1992). When a missile fails a BIT during a PRAT cycle, test personnel perform a second BIT to confirm the initial BIT failure. If the missile passes the second BIT test, testing on the missile continues and the initial BIT is designated as a Type I (BIT) false alarm. If the missile fails the second BIT test, testing on the missile continues until the end of the cycle. At the end of the cycle, test personnel conduct an additional BIT at ambient conditions. If the missile passes the ambient BIT, the second BIT is designated as a Type II (BIT) false alarm and testing continues with the next cycle. If the ambient BIT fails, the missile is subjected to a FAST in order to validate the failure. In fact, all tested missiles are subjected to a FAST after captive-carry simulation testing.

As discussed in section 1.2.3, for fielded missiles, the ground personnel normally perform a BIT before and after a sortie and the aircrew may perform several BITs during a sortie. If a BIT failure is detected, the missile is either moved to a storage area for further BIT testing or moved to another "staging aircraft platform" (if the ground personnel expect the aircraft is causing the failure indication) (Kobren, 1994). Once a missile is brought to the storage area for testing, three

consecutive BITs are taken. The missile is considered to have *failed* if any two consecutive BIT failures indicate a failed missile.

3.4 Sampling Data Constraints

Severe right censoring coupled with relatively *small* sample sizes exacerbates the estimation of $S(t)$ from the sampling data. The sampling data listed in Table 2 are *severely* right censored; on average, 73 percent of the observations in the PRAT sublot samples and 77 percent of the observations in the Operational Flight lot samples are right censored. Consequently with average sample sizes for PRAT sublots and the Operational Flight lots of 11 and 39 respectively, each sample contains few observed failures.

This sampling data constraint limits the usefulness of the non-parametric models if the captive-carry survival functions for sublots or lots are heterogeneous. The non-parametric models such as the Kaplan-Meier Survival Curve and Nelson's Cumulative Hazard Plotting model distribute mass at the failure times. With few observed failures in a sample, aggregating samples is necessary to effectively use these models. Heterogeneity prevents this aggregation. In addition to sublots or lots representing heterogeneous populations, flying regions may also represent separate populations for the Operational flight samples,; that is, $S(t)$ may also differ between flying regions.

3.5 Approach

As stated in section 1.2, the objectives of the analysis are to

- verify the current Poisson / Exponential Model and
- investigate other methods for estimating $S(t)$.
- formulate strategy for using, retiring, or refurbishing AMRAAMs.

The analysis is conducted using the parametric and non-parametric methods introduced in Chapter 2. The models are applied to both the PRAT and Operational Flight sampling data shown in Table 2. Specifically, the analysis partitions as follows:

- Examine the issue of whether the sampling data based on individual lot or sublot and flying region have arisen from identical or separate survival functions. The Mantel (1966) or log rank test and the Peto-Peto (1972) modification of the Wilcoxon test along with the Cox Proportional Hazards Model parameter tests (Kablfleish and Prentice, 1980) are used to investigate the issue.
- Verify the current estimates of the captive-carry survival functions by applying the Poisson / Exponential Model to the data sets.
- Investigate the validity of the Poisson / Exponential model by using non-parametric models. The Kaplan-Meier Survival Curve, Bootstrapping the Kaplan-Meier product limit estimator, and Nelson Cumulative Hazard Plotting model provide a means to graphically verify the exponential assumption. If the exponential assumption appears appropriate, the Lawson's extension to the Gibbs Sampling algorithm furnishes a mixed (parametric and non-parametric) estimate of the complete survival function.

The remainder of this section details the implementation of the models.

As stated in section 2.2.3, the log rank test (Mantel, 1966) and Peto-Peto (1972) modification to the Wilcoxon test provide a method to test whether two or more samples have arisen from identical survival functions. For the log rank test, the test statistic is

$$\boldsymbol{v} V^{-1} \boldsymbol{v}$$

where $\boldsymbol{v}$ is the difference between observed and "expected" failures (equation 6) and V is the covariance of $\boldsymbol{v}$ as defined in section 2.2.3. For the Peto-Peto modification to the Wilcoxon test,

the test statistic is

$$\boldsymbol{v}V^{-1}\boldsymbol{v}$$

where $\boldsymbol{v}$ is the difference between observed and "expected" failures (equation 7) and V is the covariance of $\boldsymbol{v}$ as defined in section 2.2.3. Appendix A details a S-plus function implementing the log rank test and the Peto-Peto modification to the Wilcoxon test.

The Cox Proportional Hazards Model (Cox, 1972) shown in equation 8 can be used to regress a sample indicator variable against captive-carry lifelength to determine whether two or more samples have arisen from the same survival function by testing the null hypothesis $\beta = 0$. The value $\hat{\boldsymbol{\beta}}$ that maximizes equation 9 can be obtained by a Newton-Raphson iteration utilizing equation 10 and equation 11. The large sample test, likelihood ratio test, and score test can be used to test the null hypothesis. Additionally, to check whether the large sample likelihood assumptions of the proportional hazard model are reasonably accurate, a plot of the log relative likelihood versus β

$$R(\beta) = \log L(\beta) - \log L(\hat{\beta})$$

to a plot of the normal likelihood versus β

$$-\frac{1}{2}I(\hat{\beta})(\hat{\beta} - \beta)^2.$$

can be made. If the plots show close agreement, then large sample likelihood assumptions should be reasonably accurate.

The parameter λ characterizes the Poisson / Exponential Model. As shown in section 2.2.2, when a sample is subject to independent right censoring, the maximum likelihood estimator of λ is given by:

$$\hat{\lambda} = \frac{m}{\prod_{i=1}^{n} y_i}; \tag{12}$$

An estimate of $S(t)$ is:

$$\hat{S}(t) = \exp^{(-\hat{\lambda}t)}.$$

The estimate of the mean time between failure (MTBF) or mean of S is $1/\hat{\lambda}$ and an estimate of the variance is $1/\hat{\lambda}^2$.

The Kaplan-Meier Survival Curve (Kaplan and Meier, 1958) furnishes a non-parametric estimation of $S(t)$. The curve is formed from the product limit estimate of $S(t)$; that is

$$\hat{S}(t) = \prod_{j \,|\, y^{(j)} \leq t} (1 - \frac{d_j}{n_j}) \tag{13}$$

where $y^{(j)}, j \leq m$ denotes the distinct, ordered uncensored observations (i.e., observed failure times); n_j is the number of (right censored and uncensored) observations y_i such that $y_i > y^{(j-1)}$; and d_j is the number of (right censored and uncensored) observations y_i such that $y^{(j-1)} < y_i \leq y^{(j)}$. Alternatively, n_j is the number of items *alive* just after $y^{(j-1)}$ and d_j is the number of items that have *died* in the interval just after $y^{(j-1)}$ and up to and including $y^{(j)}$. An asymptotic variance of $\hat{S}(t)$ is provided by Greenwood's formula (Greenwood, 1926); that is

$$v\hat{a}r[\hat{S}(t)] = \hat{S}^2(t) \sum_{j \,|\, y^{(j)} \leq t} \frac{d_j}{n_j(n_j - d_j)}. \tag{14}$$

Appendix A gives computer code (in the language S-plus) for implementing these estimates.

Bootstrapping the product limit estimate of $S(t)$ (Efron, 1981) provides an alternative estimate of the variance of $\hat{S}(t)$. The bootstrapping procedure is conducted as follows:

1. For K iterations, randomly sample n times, with replacement, from the observed data, $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$.
2. For each bootstrap sample, $\boldsymbol{y}^{*1}, \boldsymbol{y}^{*2}, \ldots, \boldsymbol{y}^{*K}$, calculate the product limit estimate of $S(t)$, $\hat{S}^{*1}, \hat{S}^{*2}, \ldots, \hat{S}^{*K}$, using equation 13.

3. Calculate the bootstrap estimate of the variance of $\hat{S(t)}$ as

$$v\hat{a}r[\hat{S}(t)] = \hat{S}^2(t)\frac{\sum_{j=1}^{K}(\hat{S}^{*j})^2 - (\sum_{j=1}^{K}\hat{S}^{*j})^2/K}{K-1} \tag{15}$$

Appendix A gives computer code (in the language S-plus) for implementing the bootstrap procedure. Additionally, a survival curve is constructed based on the 10, 50, and 90 percent quantiles of $\hat{S}^{*1}, \hat{S}^{*2}, \ldots, \hat{S}^{*K}$. The quantiles provide an alternate graphical analysis of variation in the product limit estimator.

As stated in section 2.2.3, Nelson's (1972) graphical procedure estimates the cumulative hazard function $H(t)$. Nelson's procedure is accomplished as follows:

1. The observations $y_1, y_2, \ldots, y_n$ are ordered with respect to magnitude (shortest time to largest time) and assigned reverse rank numbers, $y^1 \Rightarrow$ rank n, $y^2 \Rightarrow$ rank $n-1$, ..., $y^n \Rightarrow$ rank 1.

2. An estimate of the hazard rate, $h(t)$, in percent is calculated for all ordered observations $(y^i, i = 1, 2, \ldots, n)$ as

$$\hat{h}(y^i) = 100(\frac{1}{rank_i})$$

   where $rank_i$ is the rank of y^i.

3. An estimate of $H(t)$ is calculated for each y^i as

$$\hat{H}(y^i) = \sum_{j=1}^{i}\hat{h}(y^j) \tag{16}$$

4. A plot of $\hat{H}(t)$ versus t is made for the *uncensored* observations.

Two plots are made with the first on the real coordinate axes and second on a log-log scaling of the real coordinates axes. On the real coordinate axes, a plot of $\hat{H}(t)$ versus t for the exponential distribution appears as a straight line. On a log-log scaling of these axes, a plot of $\hat{H}(t)$ versus t

for the Weibull distribution appears as a straight line[3] for either an increasing or decreasing failure rate. The cumulative hazard function for a Weibull distribution is

$$H(t) = (t/\alpha)^{\beta}$$

or equivalently

$$\log(t) = (1/\beta)\log(H) + \log(\alpha).$$

The Weibull distribution describes a decreasing failure rate if $\beta < 1$; a constant failure rate (exponential distribution) if $\beta = 1$; and an increasing failure rate if $\beta > 1$. Appendix A provides computer code (in the language S-plus) implementing Nelson's procedure.

Section 2.2.3 details Lawson's (1994) Bayesian non-parametric estimator using a Gibbs Sampling algorithm. An implementation of Lawson's esimator for sampling data subject to right censoring is as follows:

1. Divide the observations $(y_i, i = 1, 2, \ldots, n)$ into a subsample of uncensored observations $(u_j, j = 1, 2, \ldots, m)$ and a subsample of right censored observations $(r_k, k = 1, 2, \ldots, c)$.
2. Give arbitrary failure times $(z_k^{(0)}, k = 1, 2, \ldots, c)$ to the right censored observations such that $\boldsymbol{z}$ is in $(\boldsymbol{r}, \infty)$.
3. For iteration $\ell, \ell = 1, 2, \ldots, K$
   (a) Generate $z_1^{(\ell)}$ such that

$$z_1^{(\ell)} \sim \begin{cases} \exp(\lambda_1^{(\ell)}) & \text{with probability } p \\ \bar{F}_1^{(\ell)} & \text{with probability } 1 - p \end{cases}$$

[3]Straight line with a non-zero intercept. Points that lie on a straight line on a zero intercept demonstrate a constant failure rate

where $\lambda_1^{(\ell)} \sim$ Gamma $(a + n_1^{\star(\ell)},\, b + \sum_{j=1}^{m} u_j^\star + \sum_{k=2}^{c} z_k^{\star(\ell-1)})$ and $\bar{F}_1^{(\ell)}$ is the empirical distribution formed from $z_k^{\star(\ell-1)}, k = 2, 3, \ldots, c$. The notation $n^{\star(\ell-1)}$ refers to the number of unique elements in $u_j, j = 1, 2, \ldots, c$ and in $z_k^{(\ell-1)}, k = 2, 3, \ldots, c$ and the notation $u_j^\star$ and $z_k^{\star(\ell-1)}$ refers to the unique elements.

(b) Generate $z_s^{(\ell)}$ such that

$$z_s^{(\ell)} \sim \begin{cases} \exp(\lambda_s^{(\ell)}) & \text{with probability } p \\ \bar{F}_s^{(\ell)} & \text{with probability } 1-p \end{cases}$$

where $\lambda_k^{(\ell)} \sim$ Gamma $(a + n_s^{\star(\ell-1)},\, b + \sum_{j=1}^{m} u_j^\star + \sum_{k=1}^{s-1} z_k^{\star(\ell)} + \sum_{k=s+1}^{c} z_k^{\star(\ell-1)})$ and $\bar{F}_s^{(\ell)}$ is the empirical distribution formed from $z_k^{\star(\ell)}, k = 1, 2, \ldots, s-1$ and $z_k^{\star(\ell-1)}, k = s+1, s+2, \ldots, c$.

(c) Generate $z_c^{(\ell)}$ such that

$$z_c^{(\ell)} \sim \begin{cases} \exp(\lambda_c^{(\ell)}) & \text{with probability } p \\ \bar{F}_c^{(\ell)} & \text{with probability } 1-p \end{cases}$$

where $\lambda_c^{(\ell)} \sim$ Gamma $(a + n_c^{\star(\ell-1)},\, b + \sum_{j=1}^{m} u_j^\star + \sum_{k=1}^{c-1} z_k^{\star(\ell)})$ and $\bar{F}_c^{(\ell)}$ is the empirical distribution formed from $z_k^{\star(\ell)}, k = 1, 2, \ldots, c-1$.

4. The result of the iterations are K uncensored samples comprised of $\boldsymbol{u}$ and $\boldsymbol{z}^{(\ell)}$.

Since the $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots \boldsymbol{z}^{(K)}$ form a Markov chain, choose every vth sample from the K uncensored samples in order to have independent samples. Appendix A provides a Fortran program implementing the above algorithm.

3.6 Summary

The chapter detailed the the methodology used for this thesis effort. Sampling data used in the analysis consists of Production Reliability Acceptance Testing and Operational Flight Data. Severe right censoring coupled with relatively small sample sizes constrains the estimation of $S(t)$ from the sampling data. The objectives of the analysis are to

- verify the current Poisson / Exponential Model
- investigate other methods for estimating $S(t)$
- formulate strategy for using and retiring, or refurbishing AMRAAMs.

The analysis is conducted using the parametric and non-parametric models introduced in Chapter 2. Appendix A provides computer code (in the language S-plus) and a Fortran program to implement the models.

IV. Findings

4.1 Introduction

This chapter documents the results of the statistical analysis of the captive-carry survival function. The chapter contains four sections. Section 4.2 (Sampling Data Modifications) describes, in general terms, the modifications made to the sampling data for the analysis. Section 4.3 (Survival Function Comparison Tests for Sampling data) details the results of the log rank test, Peto-Peto modification to the Wilcoxon test, and the coefficient tests (using the Cox Proportional Hazards Model) for the sampling data sets. These tests are used to determine if the sampling data sets have arisen from identical or separate captive-carry survival functions. Section 4.4 provides estimates of the captive-carry survival functions for the sampling data including Kaplan-Meier and Bootstrapped Kaplan-Meier Survival Curves, plots of Nelson's estimate of the cumulative hazard function, and completed (uncensored) estimates of the survival function using Lawson's Bayesion non-parametric estimator. Section 4.5 (Summary) provides a summary of the chapter.

4.2 Sampling Data Modifications

Section 1.2.1 defines *operational* captive-carry lifelength as the cumulative length of time that a missile can be captive-carried while each *Built In Test* (BIT) testable subsystem remains functional. In order to correctly estimate the operational captive-carry survival function, the analysis requires that the Production Reliability Acceptance Test (PRAT) and Operational Flight sampling data be modified / filtered. For example, PRAT criteria calls for the recording of captive-carry lifelengths to be based on the results of Fully Automated System Test (FAST) (refer to section 1.1.2) rather than BIT results[1]. In short, PRAT test results reflect captive-carry lifelengths based on FAST, while Operational Flight captive-carry lifelengths are based on BIT. The follow-

[1] The goal of PRAT is to uncover major deficiencies in production missiles. Since FAST provides a more extensive test of the the missile than BIT, FAST results are used to accept lots rather than BIT results.

ing paragraphs describe, in general terms, the modifications made to the sampling data. For a comprehensive review of the modifications, see Appendix B.

As noted above, the captive-carry lifelength observations in the PRAT sampling data are based on FAST results. In order to determine equivalent captive-carry lifelengths based on BIT results, the PRAT results are reviewed in accordance with the following criteria: a missile fails PRAT if a type II BIT failure (i.e., two consecutive BIT failures) is detected during a missile's test sequence. This revised criteria is chosen based on its closeness to the operational failure assessment; recall (refer to section 3.3) that a missile is considered to have failed if any two consecutive BIT failures indicate a failed missile. Consequently, a new PRAT sampling data set is created based on this revised criteria. For the remainder of the thesis, the terms "**FAST**" and "**BIT**" are used to distinguish, respectively, between analysis conducted on the sampling data based on FAST and the type II BIT failure criteria.

A second modification is made in the PRAT sampling data based on a differentiation between vibration/thermal test time and thermal (only) conditioning time test time. Section 3.3 explains that a PRAT cycle consists of 25 different steps totaling 313.4 minutes including 243.4 minutes of vibration/thermal test time and 70.0 minutes of thermal (only) conditioning test time. The recorded captive-carry lifelengths from the PRAT sampling data include only the 243.4 minutes of vibration/thermal test time[2]. In order to investigate the additional effect of the 70.0 minutes of thermal (only) conditioning time, new PRAT sampling data sets are created that consider a test cycle of 313.4 minutes. For the remainder of the thesis, the terms "**Vibration Only**" and "**Total Test**" are used to distinguish, respectively, between analysis conducted on sampling data based on a 243.4 minute test cycle and a 313.4 minute test cycle.

For the Operational Flight sampling data, data points are removed for missiles that had failure indications with less than 10 hours of captive-carry time *and* no incoming BIT assessment

[2]Based on operational flight tests, AMRAAM test personnel consider vibration as the major factor inducing missile failures.

performed. Since an incoming BIT assessment is not required by USAF regulation, a missile failure indication with less than 10 hours[3] of captive-carry time may be an indication of a missile delivered as non-functional as opposed to a captive-carry failure (Gug,1994). Additionally, missiles may not have the same captive-carry survival function after being repaired (for example, if missiles age). Consequently, data points for captive-carry times after missiles are repaired are removed from the Operational Flight sampling data.

4.3 Survival Function Comparison Tests for Sampling Data

The survival function comparison tests indicate:

- that the sampling data arises from different survival functions when *compared by lot* (using the BIT assessment criteria), but
- that the sampling data arises from identical survival functions for *common lots when compared by flying region.*

However, the reader should note that findings are based on observations from a limited amount of sampling data and may not be generally applicable.

Section 3.5 describes three statistical methods for testing whether two or more samples have arisen from identical survival functions: log rank test (Mantel 1966); Peto-Peto (1972) modification to the Wilcoxon Test; and the large sample test, the likelihood ratio test, and the score test (Kalbfleish and Prentice, 1980) using the Cox Proportional Hazards Model (Cox, 1972). Table 4 and Table 5, respectively, provide the FAST and BIT comparison test results for the Hughes Missile System Company (HMSC) and Raytheon Company PRAT sampling data. The results using the Vibration Only and Total Test sampling sets are identical.

For the HMSC FAST test results (refer to Tables 4), the p-values are low (ranging from 0.00475 to 0.01034), indicating that the probability of rejecting the null hypothesis (identical sur-

[3]Projected time for two 5 hour sorties.

Hughes Missile System Company

Production Lot	N	Log Rank Test Observed	Expected	$(O-E)^2/E$	Peto-Peto Mod to Wilcoxon Test Observed	Expected	$(O-E)^2/E$
Lot 2, Sublot 3	8	6	1.410	$1.495e+01$	5.0752	1.266	11.46527
Lot 3, Sublot 1	10	3	2.212	$2.805e-01$	2.7353	1.974	0.29343
Lot 3, Sublot 2	10	3	2.517	$9.248e-02$	2.6630	2.215	0.09066
Lot 4, Sublot 1	12	2	3.342	$5.391e-01$	1.7111	2.917	0.49856
Lot 4, Sublot 2	11	2	3.361	$5.512e-01$	1.7312	2.875	0.45524
Lot 5, Sublot 1	11	3	3.027	$2.366e-04$	2.7715	2.602	0.01101
Lot 5, Sublot 2	10	1	2.869	$1.217e+00$	0.7953	2.508	1.16954
Lot 6, Sublot 1	12	2	3.262	$4.881e-01$	1.7352	2.860	0.44266

Log Rank Test Statistic = 18.4 on 7 degrees of freedom, $p = 0.01034$
Peto-Peto Test Statistic = 16.3 on 7 degrees of freedom, $p = 0.02287$

Cox Proportional Hazards Model

$\hat{\beta}$	$\exp(\hat{\beta})$	$\hat{se}(\hat{\beta})$
−0.292	0.746	0.108

Large Sample Test Statistic = −2.71 on 1 degree of freedom, $p = 0.0067$
Likelihood Ratio Test Statistic = 7.97 on 1 degree of freedom, $p = 0.00475$
Score Test Statistic = 7.89 on 1 degree of freedom, $p = 0.00497$

Raytheon Company

Production Lot	N	Log Rank Test Observed	Expected	$(O-E)^2/E$	Peto-Peto Mod to Wilcoxon Test Observed	Expected	$(O-E)^2/E$
Lot 2, Sublot 3	10	3	1.768	0.8582	2.7528	1.619	0.79389
Lot 3, Sublot 1	10	3	2.320	0.1996	2.6854	2.054	0.19395
Lot 3, Sublot 2	10	0	2.425	2.4247	0.0000	2.166	2.16647
Lot 4, Sublot 1	12	4	3.378	0.1146	3.2908	2.947	0.04004
Lot 4, Sublot 2	12	1	3.412	1.7051	0.7834	2.984	1.62252
Lot 5, Sublot 1	12	2	3.084	0.3807	1.7978	2.713	0.30851
Lot 5, Sublot 2	11	4	2.894	0.4230	3.4142	2.520	0.31712
Lot 6, Sublot 1	12	5	2.721	1.9099	4.6515	2.372	2.18926

Log Rank Test Statistic = 8.1 on 7 degrees of freedom, $p = 0.3204$
Peto-Peto Test Statistic = 8.7 on 7 degrees of freedom, $p = 0.2759$

Cox Proportional Hazards Model

$\hat{\beta}$	$\exp(\hat{\beta})$	$\hat{se}(\hat{\beta})$
0.543	1.06	0.102

Large Sample Test Statistic = 0.53 on 1 degree of freedom, $p = 0.596$
Likelihood Ratio Test Statistic = 0.28 on 1 degree of freedom, $p = 0.595$
Score Test Statistic = 0.28 on 1 degree of freedom, $p = 0.596$

Table 4. PRAT - FAST Sampling Data, Comparison Test Results

vival functions for all sublot samples) when the null hypothesis is true is small. Based on these results, the conclusion can be drawn that the FAST sublot samples for the HMSC missiles arise from different survival functions. However, when the production lot 2, sublot 3 sample is removed and the remaining sublot samples tested again, the resulting p-values increase substantially (ranging from 0.299 to 0.845) indicating there is no strong evidence that these sublots arise from different survival functions. This change indicates the power of the comparison tests is low for the PRAT sampling data due to small sample sizes of the sublots. The relatively high p-values (ranging from 0.2759 to 0.596) also support this conclusion for the FAST Raytheon Company missiles. For the BIT test results (Table 5, the p-values are low to moderate for the HMSC missiles (ranging from 0.0185 to 0.1313) and the Raytheon missiles (ranging in value from 0.009312 to 0.254). This indicates mild differences among the survival functions for the sublot samples when using the BIT assessment criteria.

For the Operational Flight sampling data, survival function comparison test results indicate a moderate difference between survival functions for certain lots in the same flying regions and no difference between the survival functions for HMSC lot 3 missiles across the flying regions. Table 2 in section 3.3 shows the samples for the Operational Flight Data. Two out of the three flying regions (Saudi and Turkey) contain missiles from different lots. Table 6 provides the comparison test results for the HMSC lot 3 and lot 5 missiles in Saudi flying region. The p-values for the tests range from 0.0505 to 0.0771, indicating a mild difference between the sample survival functions for the two lots.

Table 7 provides the comparison test results for the HMSC lot 2 and lot 3 missiles in the Turkey flying region. The p-values for the tests range from 0.5039 to 0.566 indicating no significant difference between the survival functions for the two lots based on the sampling data. The flying region also contains a sampling data set of 12 HMSC lot 6 missiles. This sample is not used in the comparison tests due to the low number of captive-carry hours (less than 200 hours) accumulated by

Hughes Missile System Company

Production Lot	N	Log Rank Test Observed	Expected	$(O-E)^2/E$	Peto-Peto Mod to Wilcoxon Test Observed	Expected	$(O-E)^2/E$
Lot 2, Sublot 3	8	5	1.728	6.195204	4.388	1.524	5.383905
Lot 3, Sublot 1	10	4	2.631	0.712814	3.435	2.303	0.556115
Lot 3, Sublot 2	10	3	2.913	0.002571	2.635	2.509	0.006353
Lot 4, Sublot 1	12	3	3.642	0.113284	2.482	3.146	0.139845
Lot 4, Sublot 2	12	3	3.585	0.095542	2.703	3.045	0.038524
Lot 5, Sublot 1	11	4	3.084	0.272325	3.459	2.655	0.243640
Lot 5, Sublot 2	10	0	3.577	3.577050	0.000	3.029	3.029120
Lot 6, Sublot 1	12	3	3.840	0.183578	2.400	3.292	0.241580

Log Rank Test Statistic = 11.4 on 7 degrees of freedom, $p = 0.123$
Peto-Peto Test Statistic = 11.2 on 7 degrees of freedom, $p = 0.1313$

Cox Proportional Hazards Model

$\hat{\beta}$	$\exp(\hat{\beta})$	$\hat{se}(\hat{\beta})$
−0.221	0.802	0.0958

Large Sample Test Statistic = −2.31 on 1 degree of freedom, $p = 0.0211$
Likelihood Ratio Test Statistic = 5.56 on 1 degree of freedom, $p = 0.0183$
Score Test Statistic = 5.55 on 1 degree of freedom, $p = 0.0185$

Raytheon Company

Production Lot	N	Log Rank Test Observed	Expected	$(O-E)^2/E$	Peto-Peto Mod to Wilcoxon Test Observed	Expected	$(O-E)^2/E$
Lot 2, Sublot 3	10	7	2.222	10.2765	5.865	1.989	7.5505
Lot 3, Sublot 1	10	5	3.381	0.7756	4.054	2.815	0.5453
Lot 3, Sublot 2	10	2	3.129	0.4075	1.899	2.645	0.2106
Lot 4, Sublot 1	12	3	5.819	1.3655	2.091	4.603	1.3703
Lot 4, Sublot 2	12	3	5.008	0.8051	2.281	4.068	0.7854
Lot 5, Sublot 1	12	2	4.888	1.7063	1.730	3.972	1.2647
Lot 5, Sublot 2	11	4	4.915	0.1703	3.160	3.850	0.1237
Lot 6, Sublot 1	12	8	4.639	2.4355	6.482	3.620	2.2626

Log Rank Test Statistic = 18.7 on 7 degrees of freedom, $p = 0.009312$
Peto-Peto Test Statistic = 16.9 on 7 degrees of freedom, $p = 0.01803$

Cox Proportional Hazards Model

$\hat{\beta}$	$\exp(\hat{\beta})$	$\hat{se}(\hat{\beta})$
−0.096	0.908	0.0844

Large Sample Test Statistic = −1.14 on 1 degree of freedom, $p = 0.255$
Likelihood Ratio Test Statistic = 1.3 on 1 degree of freedom, $p = 0.254$
Score Test Statistic = 1.3 on 1 degree of freedom, $p = 0.254$

Table 5. PRAT - BIT Sampling Data, Comparison Test Results

Saudi Flying Region

		Log Rank Test			Peto-Peto Mod to Wilcoxon Test		
Production Lot	N	Observed	Expected	$(O-E)^2/E$	Observed	Expected	$(O-E)^2/E$
HMSC, Lot 3	75	28	23.687	0.7852	23.323	19.441	0.7751
HMSC, Lot 5	40	4	8.313	2.2375	3.555	7.437	2.0263

Log Rank Test Statistic = 3.4 on 1 degrees of freedom, $p = 0.06652$
Peto-Peto Test Statistic = 3.4 on 1 degrees of freedom, $p = 0.0641$

Cox Proportional Hazards Model

$\hat{\beta}$	$\exp(\hat{\beta})$	$\hat{se}(\hat{\beta})$
-0.967	0.38	0.547

Large Sample Test Statistic = -1.77 on 1 degree of freedom, $p = 0.0771$
Likelihood Ratio Test Statistic = 3.83 on 1 degree of freedom, $p = 0.0505$
Score Test Statistic = 3.37 on 1 degree of freedom, $p = 0.0665$

Table 6. HMSC Lots 3 and 5 in the Saudi Flying Region, Comparison Test Results

the missiles when compared to the samples in lots 2 and 3 and to having no uncensored observations.

Table 8 provides the comparison test results for the HMSC lot 3 missiles flown in the Italy, Saudi, and Turkey flying regions. The p-values for the tests range from 0.5039 to 0.566 indicating no evidence of a difference between the survival functions for lot 3 across the flying regions, based on the sampling data.

In summary, the PRAT BIT comparison test results indicate a mild difference between the sublot survival functions for the HMSC and Raytheon Company missiles for the sampling data. However, the power of the comparison tests is low for the PRAT sampling data due to small sample sizes of the sublots. The Saudi flying region test results for HMSC lot 3 and 5 missiles confirm this hypothesis; however, the Turkey flying region test results for HMSC lot 2 and 3 missiles indicate no significant difference between the survival functions for the sampling data. Additionally, test results for HMSC lot 3 missiles indicate no significant difference between survival functions across the Italy, Saudi, and Turkey flying regions.

Turkey Flying Region

		Log Rank Test			Peto-Peto Mod to Wilcoxon Test		
Production Lot	N	Observed	Expected	$(O-E)^2/E$	Observed	Expected	$(O-E)^2/E$
HMSC, Lot 2	34	13	11.58	0.1733	10.751	9.378	0.2011
HMSC, Lot 3	45	12	13.42	0.1496	9.918	11.291	0.1670

Log Rank Test Statistic = 0.3 on 1 degrees of freedom, $p = 0.5647$
Peto-Peto Test Statistic = 0.4 on 1 degrees of freedom, $p = 0.5039$

Cox Proportional Hazards Model

$\hat{\beta}$	$\exp(\hat{\beta})$	$\hat{se}(\hat{\beta})$
-0.233	0.792	0.405

Large Sample Test Statistic = -0.575 on 1 degree of freedom, $p = 0.566$
Likelihood Ratio Test Statistic = 0.33 on 1 degree of freedom, $p = 0.565$
Score Test Statistic = 0.33 on 1 degree of freedom, $p = 0.565$

Table 7. HMSC Lots 2 and 3 in the Turkey Flying Region, Comparison Test Results

Italy, Saudi, and Turkey Flying Regions

		Log Rank Test			*Peto-Peto Mod to Wilcoxon Test*		
Flying Region	N	Observed	Expected	$(O-E)^2/E$	Observed	Expected	$(O-E)^2/E$
Italy	29	7	6.645	0.01897	6.442	5.652	0.1104
Saudi	75	28	24.279	0.57018	22.944	19.841	0.4852
Turkey	45	12	16.076	1.03332	9.257	13.149	1.1523

Log Rank Test Statistic = 1.6 on 1 degrees of freedom, $p = 0.4432$
Peto-Peto Test Statistic = 2.1 on 1 degrees of freedom, $p = 0.3505$

Cox Proportional Hazards Model

$\hat{\beta}$	$\exp(\hat{\beta})$	$\hat{se}(\hat{\beta})$
-0.211	0.81	0.217

Large Sample Test Statistic = -0.972 on 1 degree of freedom, $p = 0.331$
Likelihood Ratio Test Statistic = 0.94 on 1 degree of freedom, $p = 0.332$
Score Test Statistic = 0.95 on 1 degree of freedom, $p = 0.33$

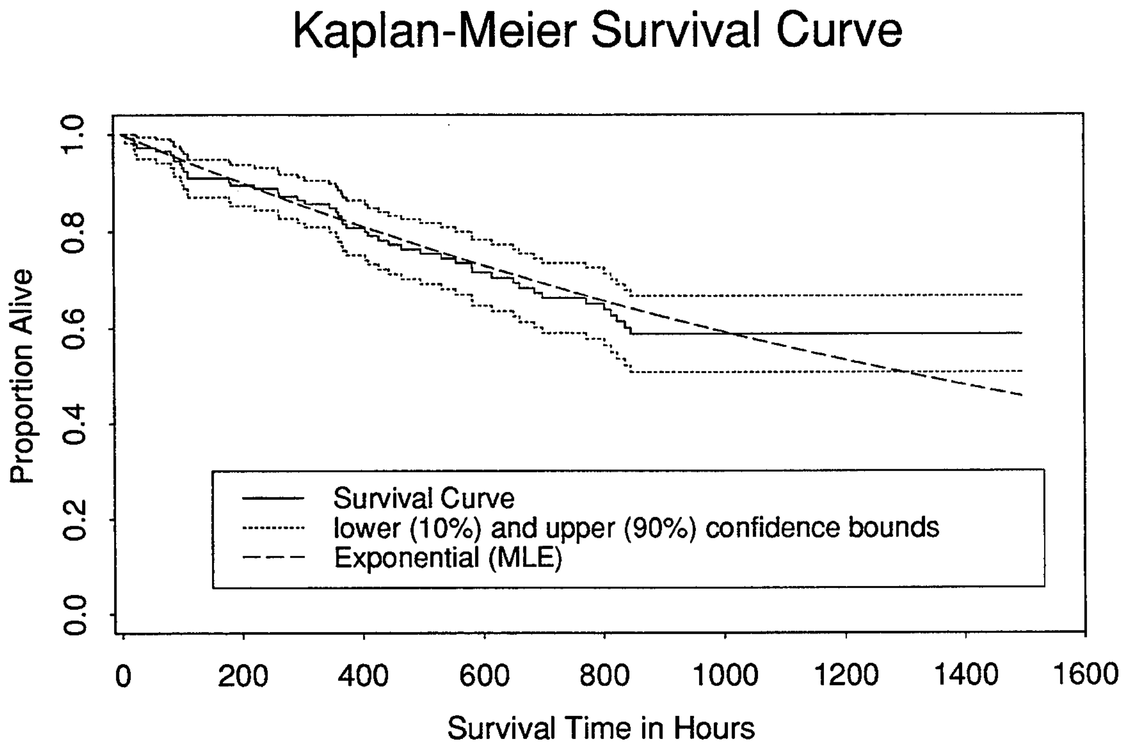
Table 8. HMSC Lot 3 in the Italy, Saudi, and Turkey Flying Regions, Comparison Test Results

4.4 Captive-Carry Survival Function

The estimated captive-carry survival functions, shown below, exhibit regions of exponential behavior (i.e., regions of constant failure rate), but the aggregate survival functions are not exponential. As detailed in section 4.3, each contractor's lot (and possibly sublot) may possess a statistically unique survival function; ideally, estimates of the survival function should be made using sampling data from the same missile lot / sublot. Unfortunately, only the Operational Flight sampling data for the HMSC lot 3 missiles possess large enough sample sizes to estimate a lot-level survival function. Further estimation requires an aggregation of lot samples. This section begins with the estimate of the captive-carry survival function for the Operational Flight HMSC lot 3 missiles, continues with an estimate of the survival function for the all Operational Flight HMSC missiles in the flying regions, and ends with an analysis of the survival functions for the HSMC and Raytheon Company PRAT missiles.

Figure 3 displays the Kaplan-Meier Survival Curve and the Bootstrapped Kaplan-Meier Survival Curve for all HMSC lot 3 missiles in the Italy, Saudi, and Turkey flying regions. The curves are based on a combined sample size of 149 observations of which 47 (32%) are uncensored. The curves are generated using the S-plus functions given in Appendix A. The 10% and 90% confidence bounds on the Kaplan-Meier Survival Curve are calculated using Greenwood's formula (equation 14). The Bootstrapped Kaplan-Meier Curve is based on 1000 bootstrap iterations with the 10%, 50%, and 90% quantiles of the uncensored observations shown. These quantiles provide a graphical analysis of the spread of the bootstrapped product limit estimators. Additionally, the calculated bootstrap variance for the uncensored observations match closely with the variance calculated using Greenwood's formula. Included in both graphs is the maximum likelihood estimate of the survival function assuming an exponential distribution. For the sampling data, $\hat{\lambda} = .0005$ failures per hour (MBTF = 1911 hours)[4]. Although the estimated exponential survival function lies within the con-

[4] 47 failures in 89802.3 hours.



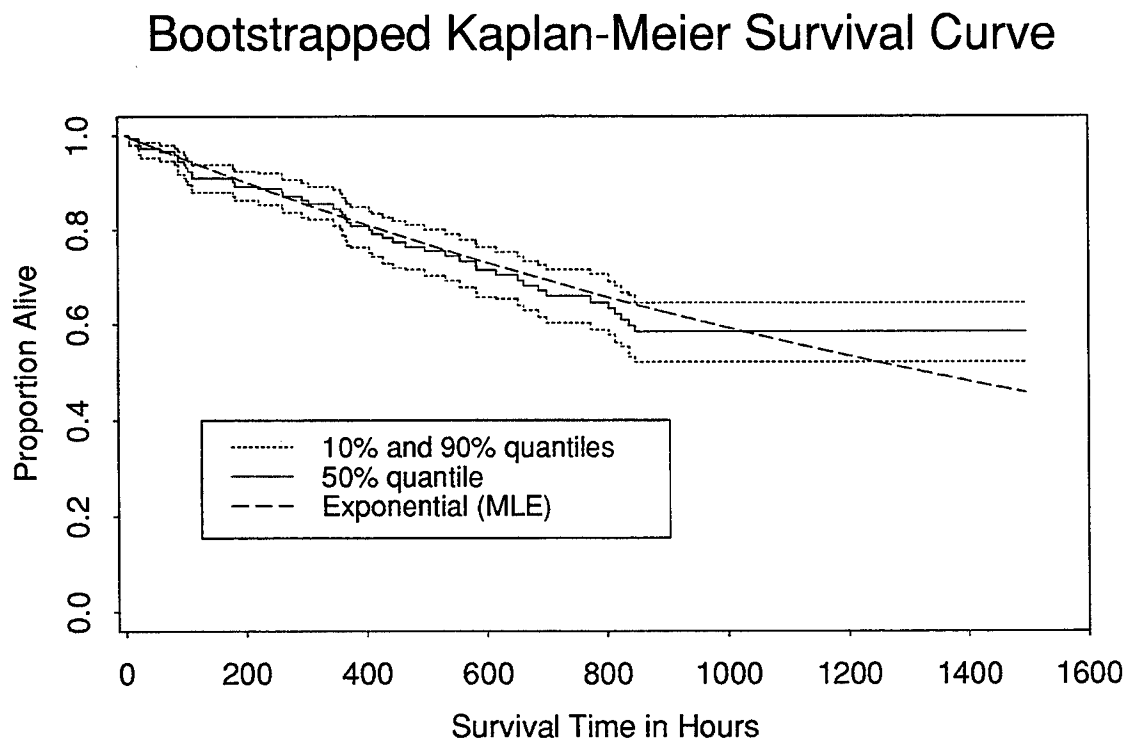


Figure 3. HMSC Lot 3 missiles (all flying regions): Kaplan-Meier Estimators

fidence bounds / outer quantiles of the non-parametric estimates, the exponential survival function deviates from the non-parametric estimates indicating deviations from a constant failure rate.

In order to investigate these deviations, Nelson's cumulation hazard plotting technique is performed. Figure 4 shows the results. The estimate of the cumulative hazard function, $\hat{H}(t)$, is plotted on the real coordinate axes and on a log-log scaling of the real coordinate axes. Recall from sections 2.2.2 and 3.5 that regions of constant failure rate are indicated by points that lie on a straight line on the real coordinate scale whereas regions of increasing failure rate (IFR) or decreasing failure rate (DFR) are indicated by points that lie on a straight line[5] on the log-log coordinate scale. Three regions of IFR are identified and are labeled in Figure 4. The sample has a high initial failure rate; approximately 10 percent of the lot 3 missiles in the sample do not reach a captive-carry lifelength of 150 hours. In addition, an increase in the failure rate is seen between 350 and 400 hours as well as beyond 800 hours. The large increase in $\hat{H}(t)$ (approximately .15) at 800 hours may be evidence of severe missile aging in this time region; however, without the presence of larger uncensored times, this hypothesis can not be confirmed.

In order to confirm the observations made for the lot 3 missiles, the same analysis is conducted for all HSMC missiles in the three flying regions. This sampling data contains 235 observations of which 64 (27%) are uncensored. Figure 5 and Figure 6 show respectively the estimated survival curves and cumulative hazard function. The reader should note that 1.) that approximately 60% of the observations are from HSMC lot 3 missiles and 2.) mild differences may exist between lot survival functions (e.g, lot 3 and lot 5 in the Saudi fly region). Each of these factors may prevent a valid estimation. For Figure 6,the Bootstrapped Kaplan-Meier Curve is also based on 1000 bootstrap iterations and the the maximum likelihood estimate of parameter of the exponential survival function, $\hat{\lambda}$, is .0004 failures per hour (MBTF = 2120 hours)[6]. As with the estimates for the lot 3 missiles,the estimated exponential survival function for the combined HSMC sample lies

[5]Straight line with a non-zero intercept. Points that lie on a straight line on a zero intercept demonstrate a constant failure rate

[6]64 failures in 135681.5 hours.

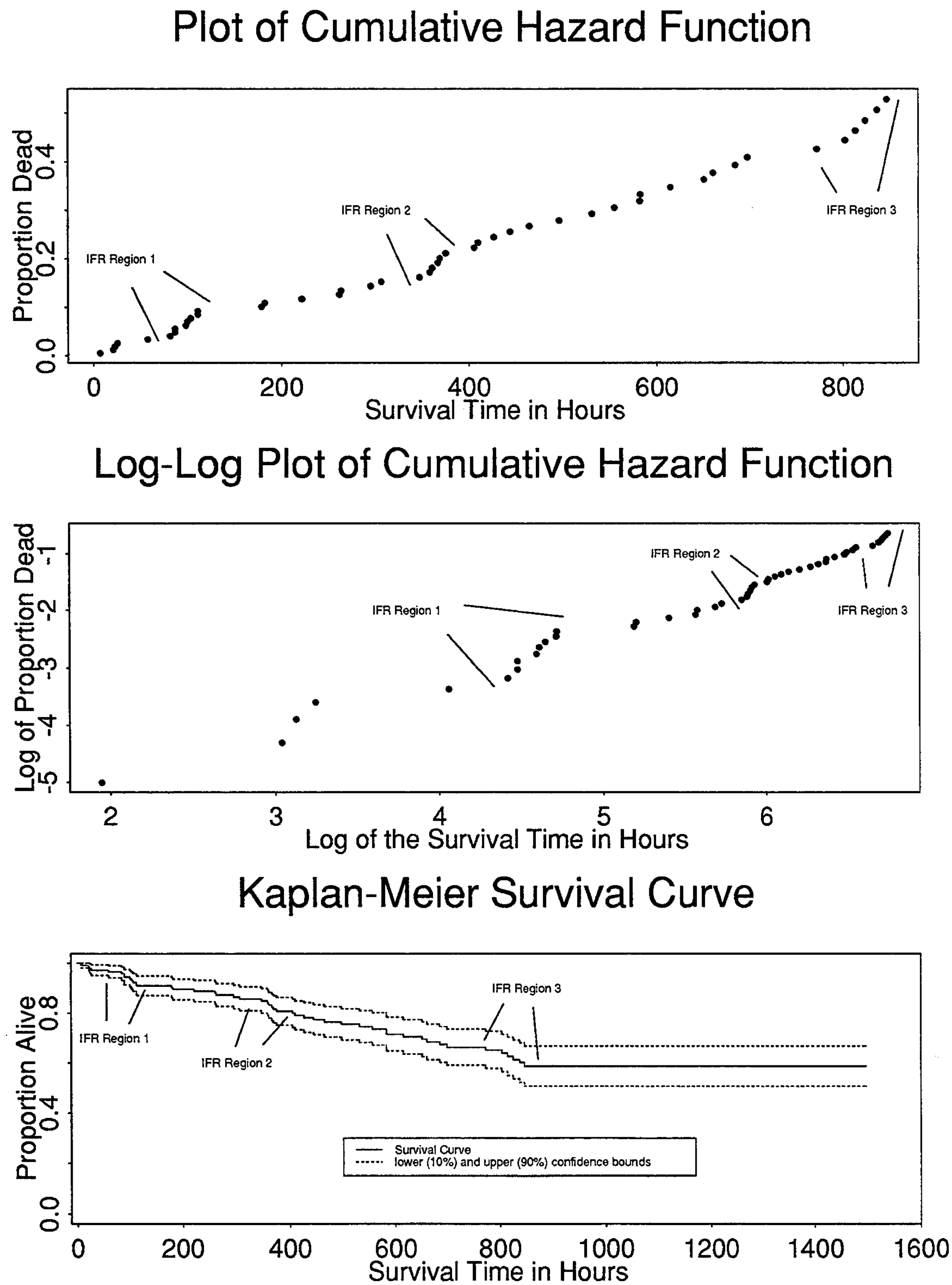


Figure 4. HMSC Lot 3 missiles (all flying regions): Nelson's estimator

within the confidence bounds / outer quantiles of the non-parametric estimates, but deviates in certain time regions from the non-parametric estimates indicating deviations from a constant failure rate.

Figure 6 shows the estimate of the cumulative hazard function. The three IFR regions identified in Figure 4 appear again. Again the sample has a high initial failure rate before 150 hours. However, the IFR region at 800 captive-carry hours ends after 900 captive-carry hours suggesting that severe aging does not take place at this captive-carry time.

In addition to the Kaplan-Meier and Bootstrapped Kaplan-Meier product limit estimators, survival curves are calculated using Lawson's non-parametric Bayesian estimator for the HMSC Lot 3 missiles in the Italy, Saudi, and Turkey flying regions as well as for all HMSC missiles in the three flying regions. These survival curves are complete (uncensored) estimates of the survival functions. Figure 7 and Figure 8 show the survival curves for the lot 3 missiles and Figure 9 shows the curves for all missiles in the flying regions. The curves are produced using the Fortran program in Appendix A. The parameters of the Gamma prior are $a = 100, b = 191100$ for the lot 3 missiles and are $a = 110, b = 233200$ for the all fly region missiles. The parameters are chosen based on graphically matching a Pareto distribution[7] with parameters a and b to the maximum likelihood estimated exponential survival curves[8] at the .1, .5, and .9 quantiles. In both cases, the algorithm is run for 1000 iterations taking observations every 10th iteration with three different values of $\alpha(\mathcal{R})$: $5, 50$, and 100[9].

Figure 7, 8, and 9 also provide a comparison of the non-parametric Bayesion estimated survival curves to the calculated Kaplan-Meier survival curves. The curves tend to match more closely with higher values of $\alpha(\mathcal{R})$. As can be seen in the Figure 7, a significant benefit of Lawson's estimator is an extended complete (uncensored) survival curve.

[7]The Pareto distribution is the prior distribution of the captive-carry lifelength.

[8]Based on converstions with the test personnel from AMRAAM IPT, the exponential estimates where thought to provide their "best guess" estimates at the .1, .5, and .9 quantiles.

[9]A higher value of $\alpha(\mathcal{R})$ means more draws from the parametric side of the mixed Dirichlet.

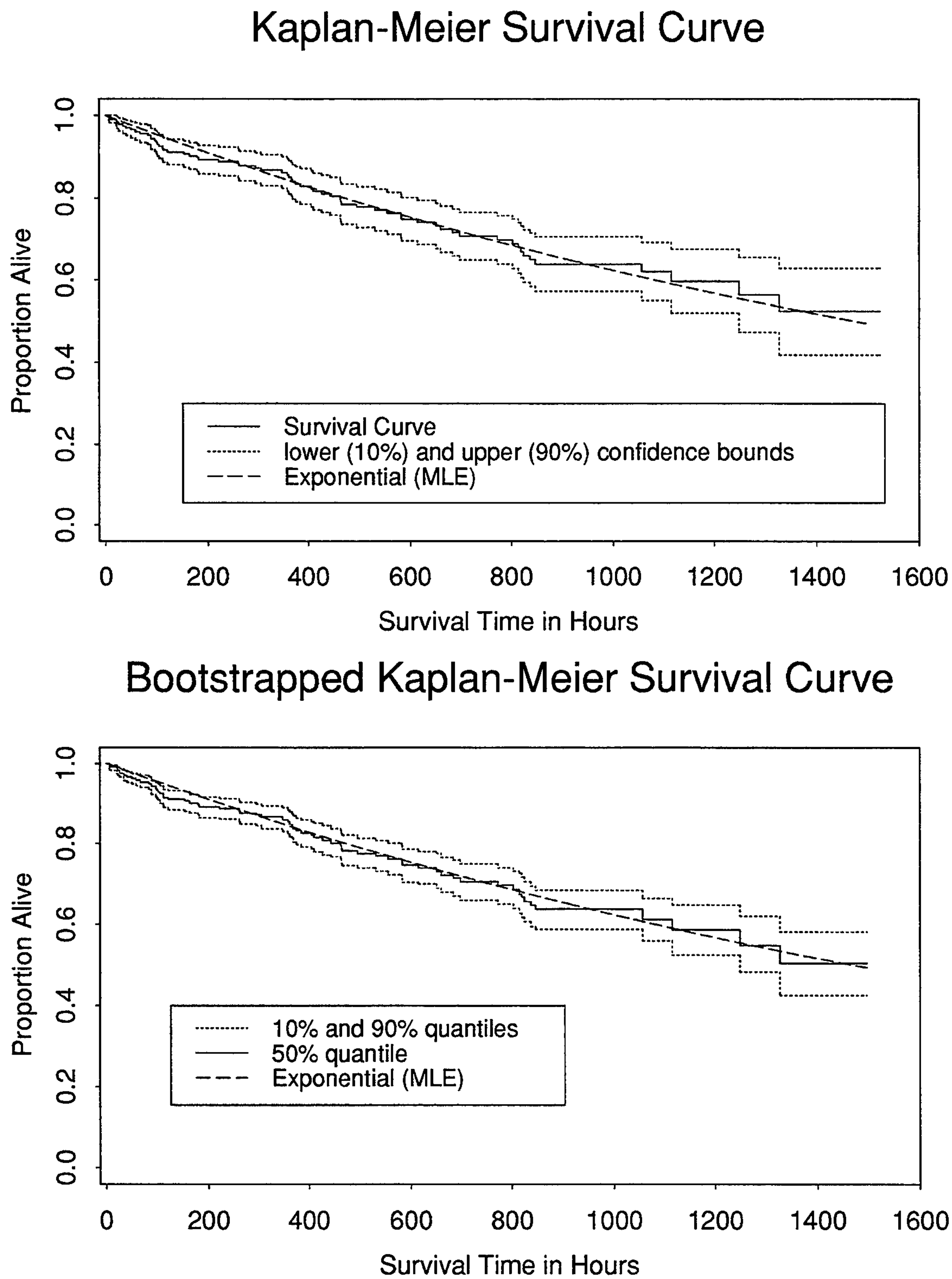


Figure 5. HMSC missiles (all flying regions): Kaplan-Meier Estimators

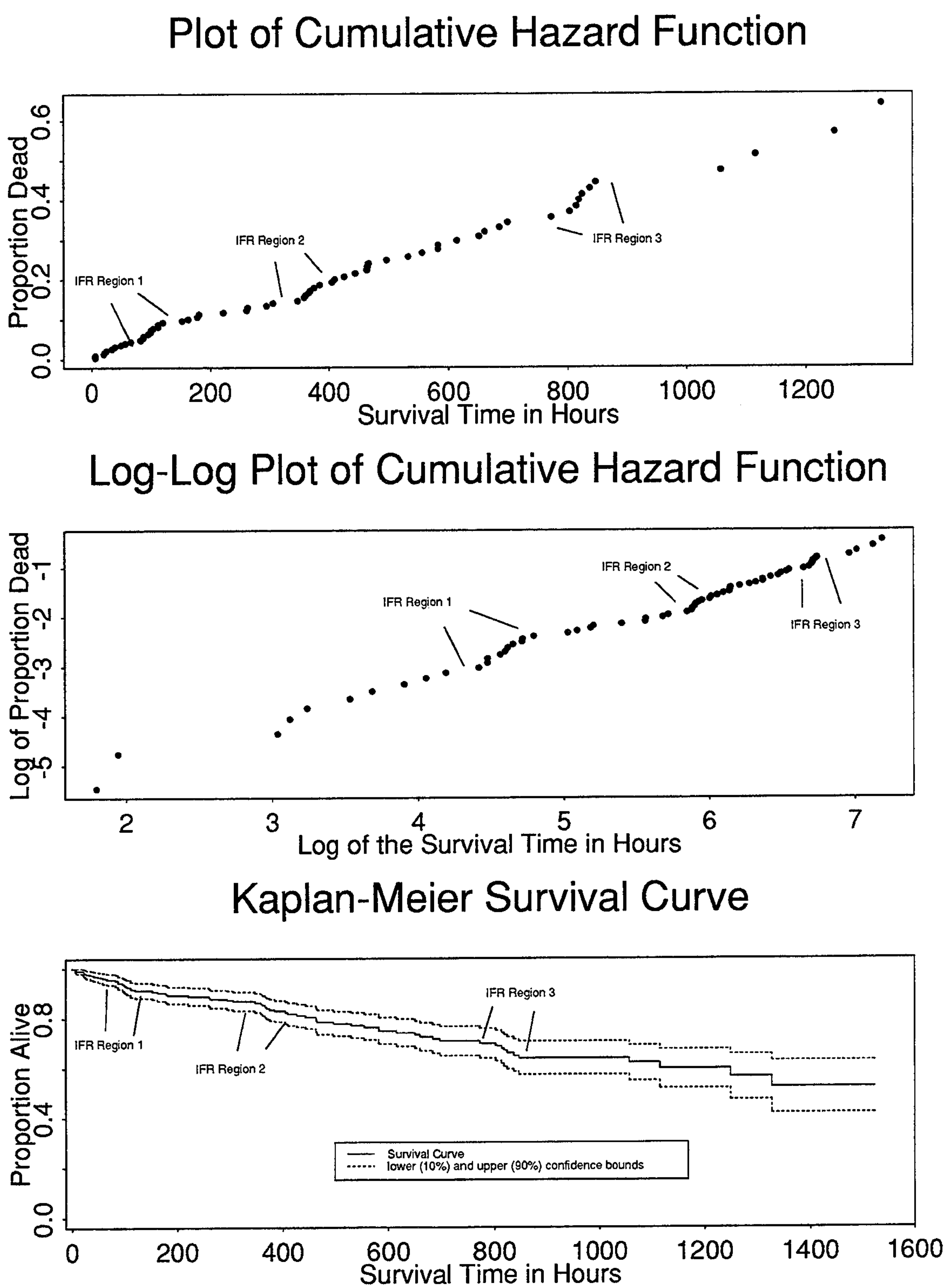


Figure 6. HMSC missiles (all flying regions): Nelson's Estimator

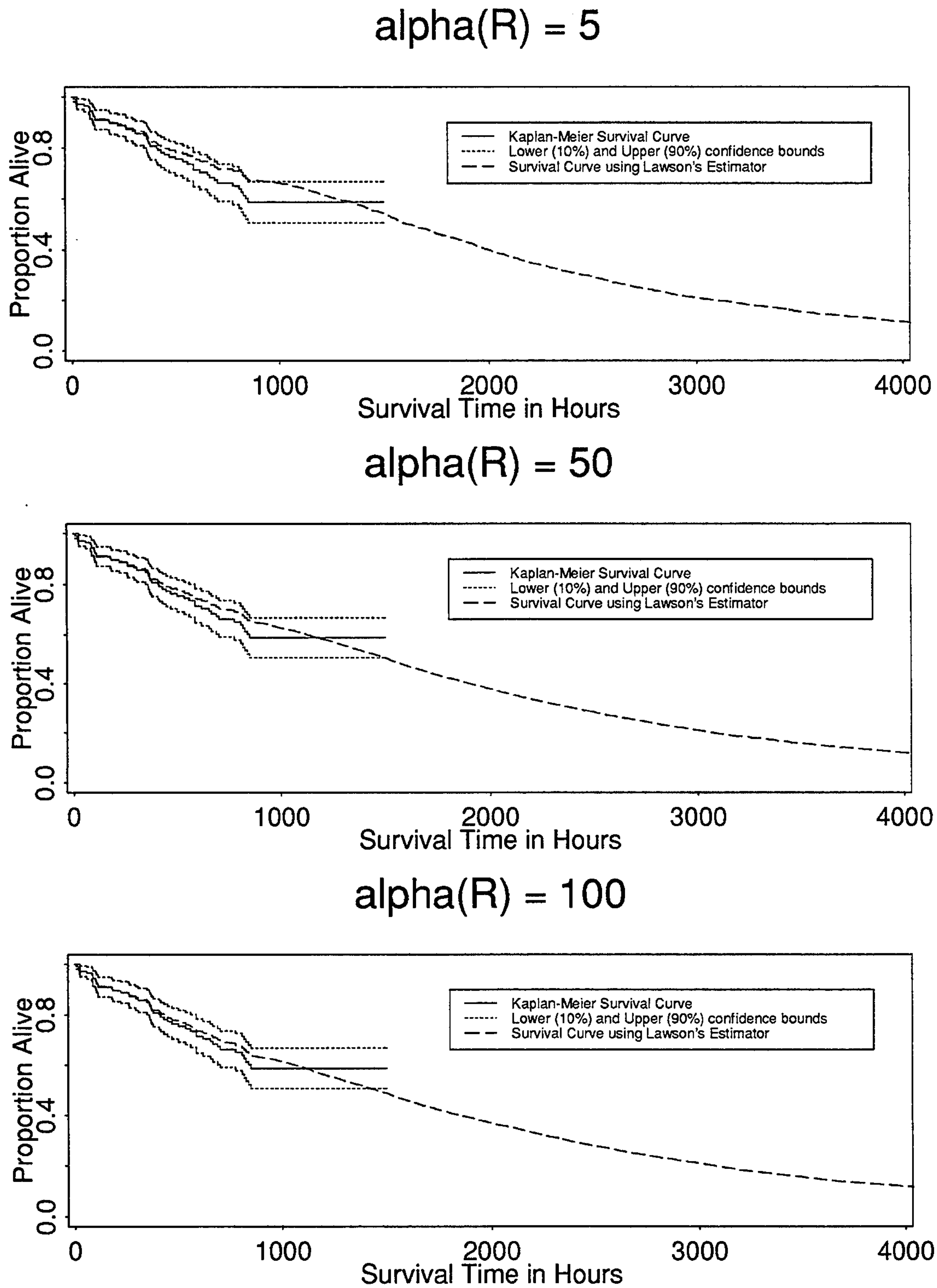


Figure 7. HMSC Lot 3 missiles (all flying regions): Lawson's Estimator

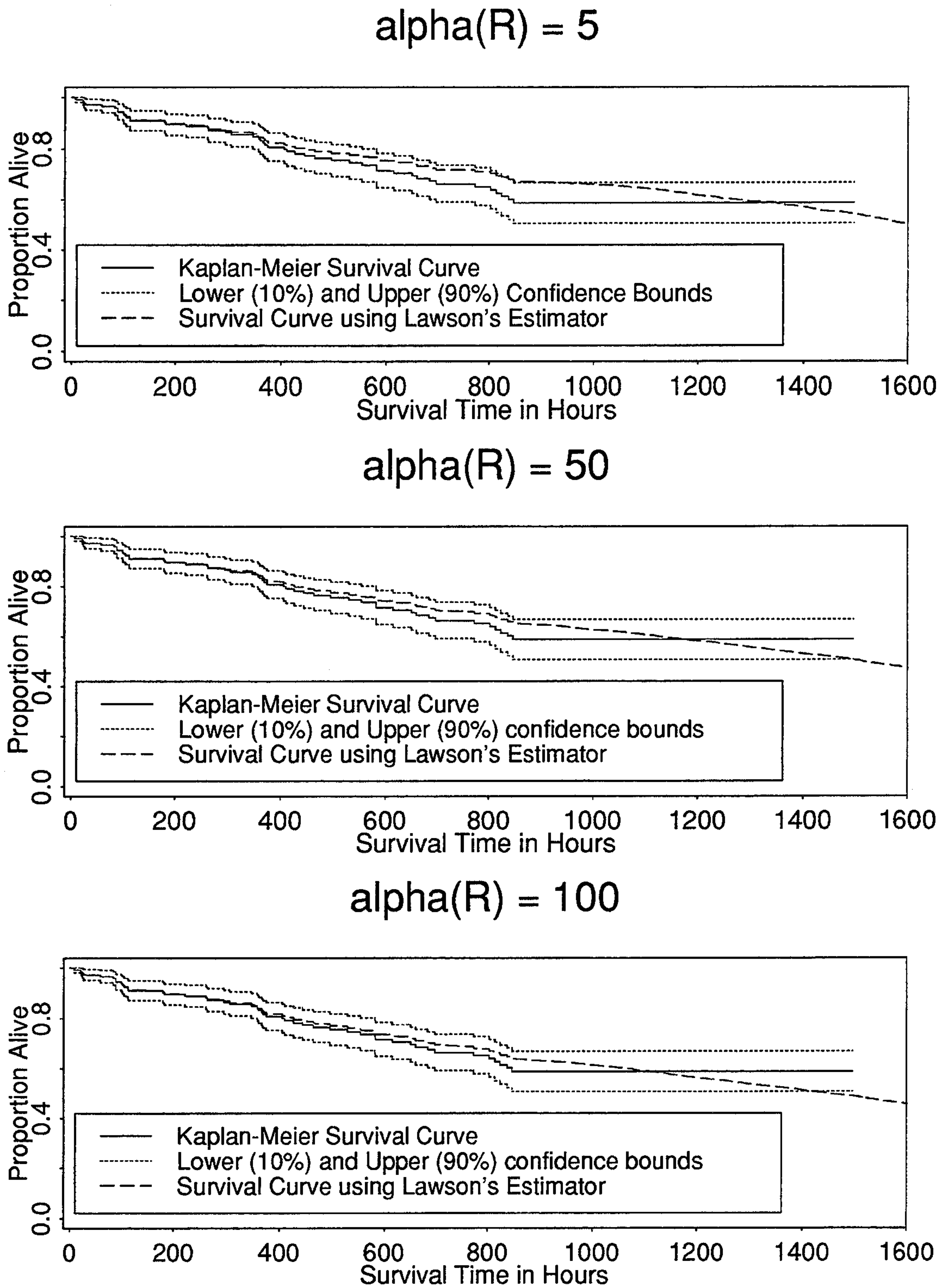


Figure 8. HMSC Lot 3 missiles (all flying regions): Lawson's Estimator

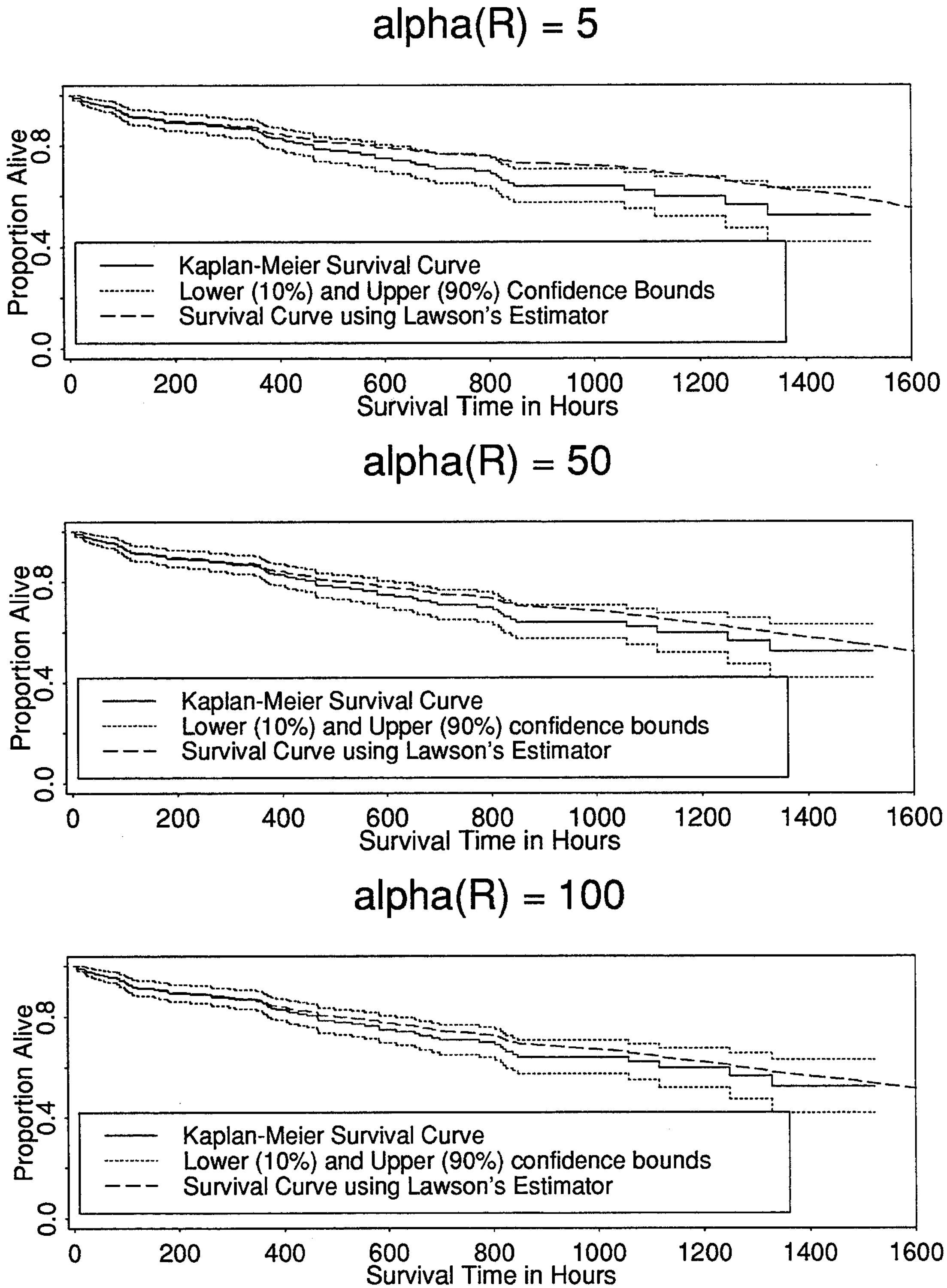


Figure 9. HMSC missiles (all flying regions): Lawson's Estimator

A comparison of the HMSC PRAT BIT (vibration time only) sampling data and HSMC Operational Flight sampling data shows significantly different initial failure rates possibly indicating the missiles will have different survival functions if used in different flying environments. Figure 10 shows the Kaplan-Meier Survival Curves and Cumulative Hazard Function Plots for HMSC PRAT and Operational Flight sampling data. Approximately 30% of the HMSC PRAT sampling data fails by 150 captive-captive hours compared with only 10% of the HMSC Operational Flight sample. In a discussion with two subject matter experts in the AMRAAM JSPO, it is believed that the PRAT flying environment is considerably more demanding on the missile reliability than the present operational flying environment[10](Guglielmoni, 1994 and Kobren, 1994).

As a final note, the analysis indicates no difference in the shape of the survival functions for the PRAT sampling data as related to the use of vibration only test time and total test time. Figure 11 shows the survival curves and cumulative hazard functions for the Raytheon Company PRAT FAST vibration only and total test sampling data. The graphs are identical in shape except for the scales on the time axes. This outcome is related to the fact that failures are recorded at approximately the same percentage of time in a cycle. For instance, the first bit .

4.5 Summary

The chapter documented the results of the statistical analysis of the captive-carry survival function. The PRAT BIT comparison test results indicated a mild difference between the sublot sample survival functions for the HMSC and Raytheon Company missiles. The Saudi flying region test results for HMSC lot 3 and 5 missiles confirmed this hypothesis; however, the Turkey flying region test results for HMSC lot 2 and 3 missiles indicated no significant difference between the sample survival functions. Additionally, test results for HMSC lot 3 missiles indicated no significant difference between sample survival functions across the Italy, Saudi, and Turkey flying regions.

[10]The PRAT environment includes several violent air-to-air combat maneuvers such as the *wind-up* turn where as the majority of the missions in the fly regions are non-combative, e.g, patrolling.

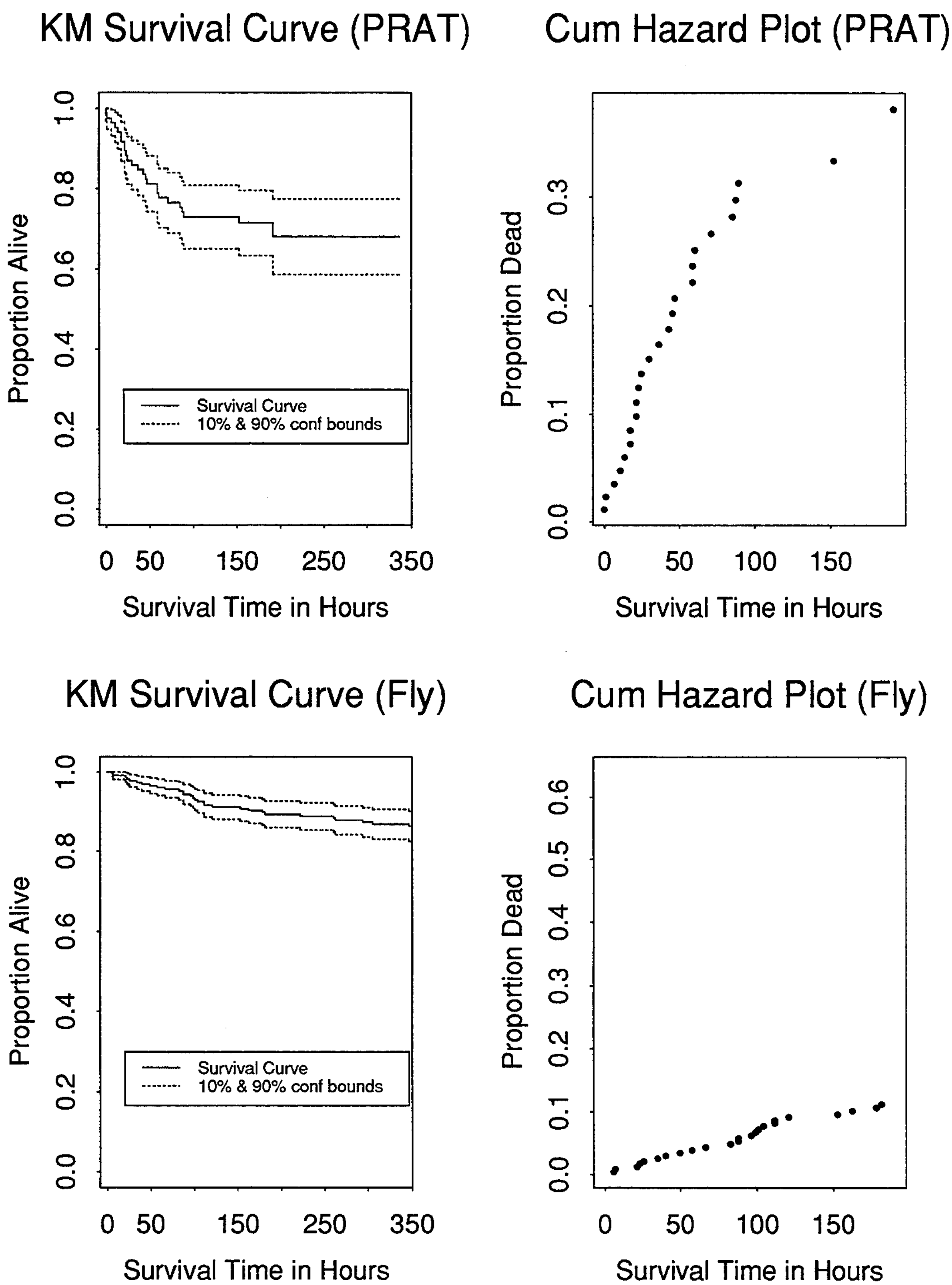


Figure 10. Comparison of HMSC PRAT BIT (Vibration) and Fly Data

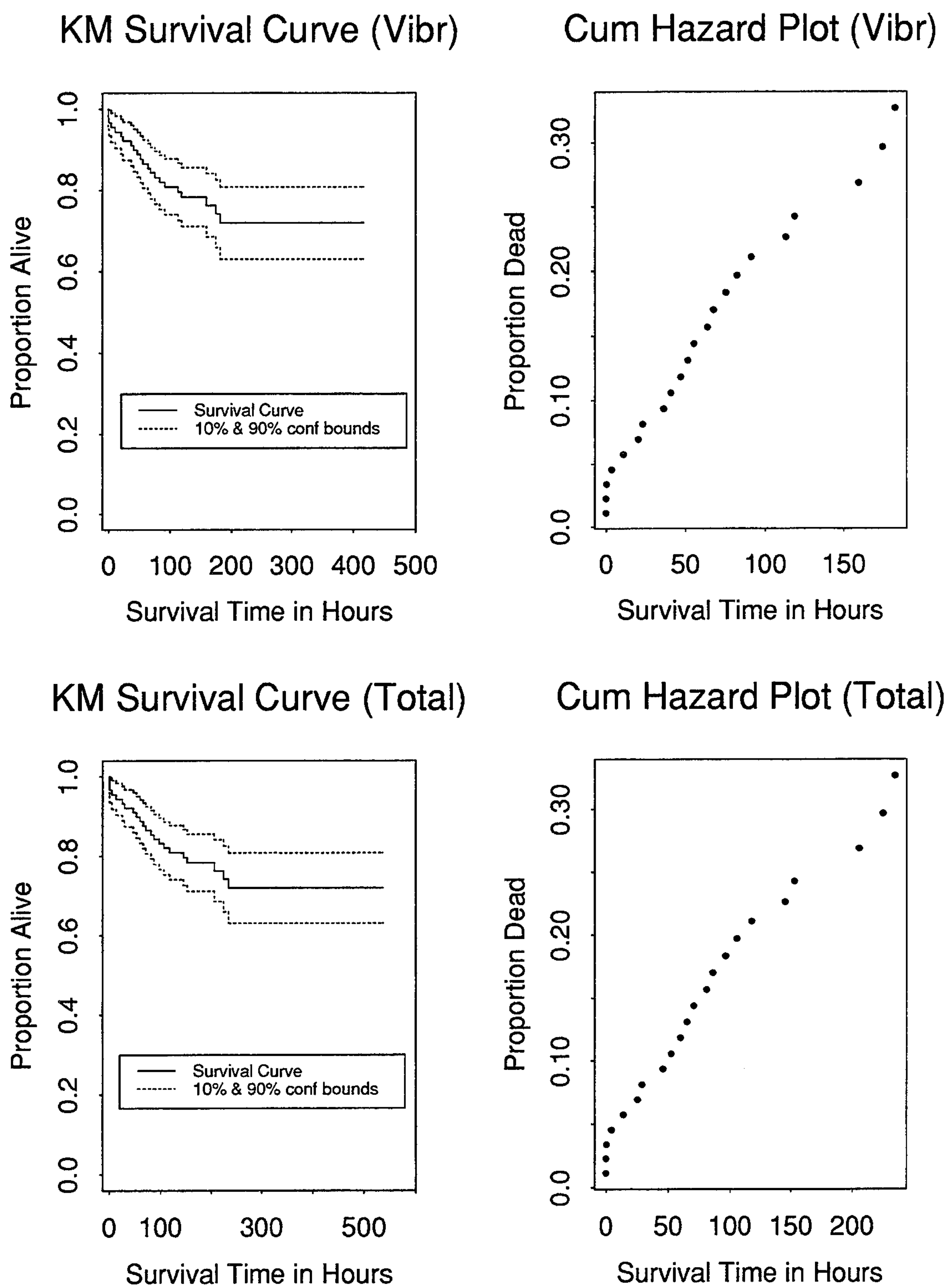


Figure 11. Comparison of Raytheon PRAT FAST Vibration and Total Time

The estimated captive-carry survival functions exhibited regions of exponential behavior (i.e., regions of constant failure rate), but the aggregate survival functions were not exponential. An estimate of the captive-carry survival function for the Operational Flight HMSC lot 3 missiles was made. Although the estimated exponential survival function ($\hat{\lambda} = .0005$ failures per hour) lies within the confidence bounds / outer quantiles of the Kaplan-Meier and Bootstrapped Kaplan-Meier estimates, the exponential survival function deviates from the non-parametric estimates indicating deviations from a constant failure rate. In order to investigate these deviations, Nelson's cumulation hazard plotting technique was performed. Nelson's plot indicated three regions of increased failure rate: before 150 hours, between 350 and 400 hours, and starting at 800 hours.

In order to confirm the observations made for the lot 3 missiles, the same analysis was conducted for all HSMC missiles in the three flying regions. As with the estimates for the lot 3 missiles,the estimated exponential survival function for the combined HSMC sample lied within the confidence bounds / outer quantiles of the non-parametric estimates, but deviated in certain time regions from the non-parametric estimates. The three IFR regions identified in HMSC lot 3 sampling data appear in the estimates.

In addition to the Kaplan-Meier and Bootstrapped Kaplan-Meier Survival Curves, survival curves were calculated using Lawson's non-parametric Bayesian estimator for the HMSC Lot 3 missiles in the Italy, Saudi, and Turkey flying regions as well as for all HMSC missiles in the three flying regions. These curves were then compared to the calculated Kaplan-Meier Survival Curves. Both curves tend to match more closely with higher values of $\alpha(\mathcal{R})$. As seen in the figures, a significant benefit of Lawson's estimator is an extended complete (uncensored) survival curve.

The chapter concludes with a comparison of the HMSC PRAT BIT (vibration time only) sampling data and HSMC Operational Flight sampling data and the Raytheon Company PRAT FAST vibration only and total test sampling data. The HMSC PRAT BIT sampling data and HSMC Operational Flight sampling data showed significantly different initial failure rates, possibly

indicating that missiles will have different survival functions if used in flying environments which differ with respect to the operational mission. The shape of the survival curves and cumulative hazard functions indicated no difference in the survival functions based on vibration only test time and total test time for the PRAT sampling data.

V. Conclusion

5.1 Introduction

This thesis considered the problem of estimating the survival function of an item from sampling data subject to partial right censoring. In particular, the case of estimating the captive-carry survival function of the AIM-120A Advanced Medium Range Air-to-Air Missile (AMRAAM) was considered. As stated in section 1.2, the thesis had three objectives:

1. verification of the current captive-carry survival function (i.e., Poisson / Exponential model);
2. investigation of other methods for estimating the captive-carry survival function;
3. formulation of a strategy for using and retiring or refurbishing AMRAAMs.

This chapter contains three sections. Section 5.2 (Taxonomy) addresses the second thesis objective by providing a general taxonomy for estimating the survival function of an item from sampling data subject to partial right censoring. Section 5.3 (AMRAAM Summary) addresses the first and third thesis objectives by recommending several missile operating procedures based on the statistical analysis of the captive-carry sampling data. Section 5.4 (Further Research) outlines the additional research necessary to fully understand the captive-carry survival function(s) for the AMRAAM.

5.2 Taxonomy

As shown in section 2.2.1 when sampling data is subject to partial right censoring, complete item lifelengths are only observable for the failed items. Consequently, when estimating the survival function of an item from sampling data subject to partial right censoring, the analyst is confronted with three choices. He can treat right censored observations as uncensored observations (i.e., item failures), remove the right censored observations from the sample, or consider the partial lifelength information provided by the right censored observations. The first two choices bias the survival function estimate. With regards to the first choice, the survival function estimate is biased low since

items with right censored lifelengths function (at least) beyond their right censored times. As to the second choice, the survival function estimate is biased with the direction of the bias indeterminant; eliminating the right censored observations removes lifelength information from consideration in the survival function estimate. The degree of bias introduced by the first two analytical choices depends on the number of right censored observations and the specific information contained in each right censored observation. Alternatively, Chapter 2 introduced statistical models that consider the lifelength information contained in right censored observations when generating a survival function estimate and Chapter 3 provides algorithms implementing these models. The models form a robust set of investigative tools for estimating the survival function of an item.

A general taxonomy for estimating the survival function of an item from sampling data subject to partial right censoring proceeds as follows:

- (when using more than one sampling data set) examine the issue of whether the sampling data sets have arisen from identical or separate survival functions. The Mantel (1966) or log rank test and the Peto-Peto (1972) modification of the Wilcoxon test along with the Cox Proportional Hazards Model parameter tests (Kablfleish and Prentice, 1980) can be used to accomplish this task.
- investigate the characteristics of the survival function by using non-parametric estimates. The non-parametric models included the Kaplan-Meier Survival Curve (Kaplan-Meier, 1958), the Bootstrapped Kaplan-Meier Product Survival Curve (Efron, 1981), Nelson's (1972) Cumulative Hazard Plotting technique, and Lawson's (1994) Bayesian estimator using Gibbs Sampling algorithm (Casella and George, 1990). These models provide a graphical means in with which to evaluate the survival function, or they can be used as the estimated survival function.

- (if appropriate) derive a parametric model (e.g. Exponential or Weibull) for the survival function or a region of the survival function using parametric estimators (e.g. maximum likelihood) [1].

The Kaplan-Meier and Bootstrapped Kaplan-Meier Survival Curves model directly the captive-carry survival function $S(t)$. Regions of increasing / decreasing failure rate are indicated by sharp / flat declines in the proportion of missiles alive. Nelson's technique estimates the cumulative hazard function $H(t)$ and as consequence of the basic relation $H(t) = -\log S(t)$ also estimates the captive-carry survival function. Distribution parameters (e.g., for the Exponential and Weibull distributions) may be estimated directly from the plots.

The usefulness of the survival curves and Nelson's plotting technique lies in the fact that the models can be used to derive, an appropriate parametric model for the survival function or (more appropriately in this case) a region of the survival function without assuming an underlying distribution. For instance, the models allow estimation of the parameters of the exponential for the constant failure rate regions as well as the parameters of the Weibull distribution for the increasing / decreasing failure rate regions for the PRAT and Operation Flight sampling data. This application is analogous to the use of histograms and quantile plots predicting a parametric model when the data is completely uncensored.

Lawson's non-parametric Bayesian estimator provides a complete, uncensored estimate of the survival function by generating uncensored observations from the right censored observation using a mixture of Dirichlets. The algorithm used in the thesis assumed an Exponential parametric family. However, the algorithm can be modified to include other parametric families such as the Weibull.

[1] A good reference for parametric models is Cohen (1991).

5.3 AMRAAM Summary

With regard to the first objective, the findings (Chapter 4) show that the estimated captive-carry survival function for the AMRAAM exhibits regions of exponential behavior (i.e., constant failure rate), but the survival function is not entirely exponential. Although assuming an exponential distribution for the captive-carry survival function provides for relatively accurate prediction of missile survival in the aggregate, the exponential assumption overestimates/underestimates missile survival in critical time regions. For instance, the exponential overestimates initial missile survival (missiles with less 150 captive-carry hours). Alternatively, the non-parametric models investigated as part of the second objective provide methods with which to observe deviations from constant failure rate.

In addition, the findings (Chapter 4) show that a mild difference exists between the captive-carry survival function of sampling data between HSMC lot 3 and 5 missiles (Saudi flying region); however, the survival functions of sampling data between HSMC lot 2 and 3 missiles (Turkey flying region) are not significantly different. These results may indicate that certain production lots can be treated as a single homogeneous population where as other lot combinations are from heterogeneous populations (refer to section 5.4 for further information).

The third thesis objective can be addressed only partially with a complete analysis requiring further research. The strategy for using AMRAAMs is discussed first and a strategy for retiring or refurbishing AMRAAMs follows. The findings (Chapter 4) show the AMRAAM to have a relatively high initial failure rate; approximately 10% of the missiles in the Operational Flight sampling data fail before 150 captive-carry hours. The initial failure rate is even higher for the PRAT HMSC missiles with the approximately 30% of the missiles failing before 150 hours. These results suggest that newly fielded missiles experience a "shake-out" period where the robustness of a missile's reliability is tested; that is, if a missile has a severe reliability problem, this problem will most likely surface in early captive-carry hours. Based on this observation, aircraft likely to be involved

in combat should not fly with missiles with less than 150 captive-carry hours. Additionally, the estimate of the survival function for the Operational Flight sampling data indicates that another significant increase in the failure rate occurs at 800 hours with approximately 10% of the sample failing between 800 and 900 hours. Whether or not this increase in failure rate persists after 900 hours can not currently be determined; only four failures are observed after 900 hours. However, the reader should note that 44 of the 235 missiles in the Operational Flight sampling data have recorded at least 1000 captive-carry hours, suggesting that the increased failure rate may not persist.

Formulation of a strategy for retiring or refurbishing AMRAAMs would require further research. The findings (Chapter 4) show that the captive-carry survival function does not always exhibit constant failure rate; in fact three periods of significant increase in the failure rate are found in the Operational Flight sampling data. Thus, the missiles do age. However, at exactly what range of captive-carry time aging becomes detrimental to missile performance is still an open issue. More data must be observed after the 1400 hour captive-carry point in order to provide a reliable indication. Hence, until more data is collected, the recommendation would be to keep flying a missile until it fails with the caveat that aircraft likely to be involved in combat situations should not be equipped solely with the missiles that have accumulated more than 1000 captive-carry hours.

With respect to the refurbishing issue, initial data is inconclusive on the resultant captive-carry survival function for repaired missiles due to the small sample size of the repaired population[2]. Most of the failures occur in the guidance section of the missile where the electronic components are housed. Current USAF maintenance policy calls for a maintenance pipeline to be established to replace failed guidance sections with repaired guidance sections from previous missile failures. The issue that this policy raises is whether all repaired guidance sections exhibit the same captive-carry survival function. For instance, if the guidance section of a missile with 300 captive-carry hours is replaced with a repaired guidance section that has accumulated 1000 hours, it may not be

[2]only seven missiles have been repaired in the Operational Flight data.

reasonable to assume that the same reliability performance be expected. Since the findings show that missiles age, the expectation is that a difference would exist. If a difference exists, significant analytical work is needed to determine when it becomes cost-effective to replace the failed guidance section with a new guidance section instead of repairing the failed guidance section. These issues will have to be addressed by further research as data becomes available.

5.4 Further Research

Suggestions for further research are:

- continue to track the operational flight data in order to accurately estimate the captive-carry survival function beyond the 1000 hour captive-carry time region,
- estimate the captive-carry survival function(s) for repaired missilesas more data becomes available,
- analyze other operational flight data sets to determine whether or not different production lots can be modeled using identical captive-carry survival functions, and
- investigate a more complete captive-carry survival function for the PRAT missiles.

Using the models documented in this thesis, AMRAAM personnel can estimate the captive-carry survival function from any sampling data set. Analysis on the Operational Flight data should continue to determine regions of increased failure rate above 1000 captive-carry hours and decide whether or not these regions are severe enough to warrant retirement or refurbishment of missiles.

For the second suggestion, the estimation of the captive-carry survival function(s) for repaired missiles must wait until more data becomes available on the captive-carry lifelength of these missiles. If the current maintenance repair policy remains in effect, the guidance sections as well as the missiles will have to be tracked. A maintenance database is available at Warner Robbins AFB that can provide the required information. Since the underlying captive-carry survival function may

be dependent on the time at which the missile first failed, a model such as the Cox Proportional Hazards Model that enables consideration of the prior captive-captive lifelengths will have to be used.

The issue of whether different production lots can be modeled using the same captive-carry survival function merits further research. The log rank test, Peto-Peto modification to the Wilcoxon test, and the coefficient tests using the Cox Proportional Hazards model provide methods to analyze this issue on other operational flight sampling data sets. Research on this issue is important since all the models discussed in this thesis except the Cox Proportional Hazards Model consider only a single survival function.

Finally, the current captive-carry survival functions for the PRAT data are only estimated to the 350 to 500 hour captive-carry time range. The findings (Chapter 4) show that the PRAT environment is more demanding on missile reliability than the current operational flight environment. If aircraft are flown in environments similiar to PRAT (for instance, in extended air-to-air combat missions), the estimated PRAT captive-carry survival functions would be extremely useful. Assuming that lots have identical survival functions, this estimate can be accomplished by extending the testing of two missiles in every sublot to failure or a reasonable time such as 1000 captive-carry hours.

Appendix A. Computer Programs

This appendix includes the computer programs used in the analysis effort. All model algorithms *except* the Lawson's non-parametric bayesion estimator[1] are written in the S–PLUS programming language. S-Plus is a software system for data analysis managed by Mathsoft, Inc.

A.1 Kaplan-Meier Product Limit Estimator

The Kaplan-Meier product limit estimates are calculated using the built-in function *surv.fit* and survival curves are generated using the built-in function *plot.surv.fit*.

Code for SURV.FIT:

```
function(time, status, strata = rep(1, length(time)), na.strata = F, type =
        "kaplan-meier", error = "greenwood", conf.level = 0.95, conf.type =
        "log", wt = rep(1, length(time)), coxreg.list, x, predict.at, center)
{
        strata <- unclass(strata)
        status <- unclass(status)
        method <- charmatch(type, c("kaplan-meier", "fleming-harrington"),
                nomatch = NA)
        if(is.na(method))
                stop("Invalid value for 'type'")
        error.int <- charmatch(error, c("greenwood", "tsiatis", "cox"), nomatch
                 = NA)
        if(is.na(error.int))
                stop("Invalid value for 'error'")
        if(!missing(conf.level))
```

[1]Lawson's estimator is coded in FORTRAN77

```
            if((length(conf.level) != 1) || !is.finite(conf.level) || (
                    conf.level <= 0) || (conf.level >= 1))
                    stop("argument 'conf.level' must be a single number
 greater than zero and less than one.")
        conf.type <- charmatch(conf.type, c("none", "plain", "log", "log-log")
, nomatch = NA)
        if(is.na(conf.type))
                stop("Invalid value for 'conf.type'")
        if(!missing(coxreg.list)) {
                if(is.null(coxreg.list$coef) | is.null(coxreg.list$var))
                        stop("'coxreg.list' must be an output list from coxreg"
                                )
                if(missing(x))
                        stop("'x' is required when 'coxreg.list' is given")
                x <- as.matrix(x)
                n <- nrow(x)
                nvar <- ncol(x)
                error.int <- 3  #Cox error.  See coxreg helpfile.
                if(missing(type))
                        method <- 2
                if(length(coxreg.list$coef) != nvar)
                        stop("Number of variables in 'coxreg.list' does not
match 'x' matrix")
        }
        else {
                n <- length(time)
```

```
                if(error.int == 3)
                        stop("Cox error specified, need 'coxreg.list' and 'x'")
        }
        if(!missing(strata))
                strata.name <- deparse(substitute(strata))
        if(is.category(strata))
                strata.level <- levels(strata)
        else {
                strata.level <- sort(unique(strata))
                strata <- category(strata, levels = strata.level)
        }
        if(na.strata && any(is.na(strata))) {
                strata[is.na(strata)] <- max(strata[!is.na(strata)]) + 1
                strata.level <- c(strata.level, NA)
        }
        if(length(status) != n || length(time) != n)
                stop("No. of observations in 'time' and 'status' must match")
        if(length(wt) != n)
                stop("'wt' vector is the wrong length")
        if(length(strata) != n) stop("'strata' vector is the wrong length")
# find observations with missing values
#   and check for legal time and status values
        nomiss <- !(is.na(time) | is.na(status) | is.na(wt) | is.na(strata))
        if(!missing(coxreg.list))
                nomiss <- nomiss & !(is.na(x %*% rep(1, nvar)))
        nused <- sum(nomiss)
```

```
        if(any(time[nomiss] < 0))

                stop("Time values must be >= 0")

        zz <- status[nomiss]

        if(any(zz > 1)) {

                zz <- zz - 1

                status[nomiss] <- zz

        }

        if(any(zz != 0 & zz != 1)) stop("Invalid status value") #

# Sort the data (or rather, get a list of sorted indices)

        sorted <- ((1:n)[nomiss])[order(strata[nomiss], time[nomiss])]

        if(!missing(coxreg.list)) {

                x <- as.matrix(x[sorted, ])    #

# The "center" arg is historical, it was a very bad label for "predict.at"

                if(missing(predict.at) & !missing(center))

                        predict.at <- center

                if(missing(predict.at))

                        predict.at <- apply(x, 2, mean)

                x <- sweep(x, 2, predict.at)

                wt <- exp(x %*% coxreg.list$coef)

                storage.mode(x) <- "double"

        }

        else {

# create some dummys

                x <- as.matrix(double(nused))

                nvar <- 1

                coxreg.list <- list(var = double(1))
```

```
            wt <- wt[sorted]
        }
        stime <- as.double(time[sorted])
        sstat <- as.integer(status[sorted])
        strata <- strata[sorted]
        if(max(strata) == 1) {
# only one group
                surv <- .C("survfit",
                        as.integer(nused),
                        as.integer(nvar),
                        time = stime,
                        sstat,
                        as.double(wt),
                        as.integer(method),
                        as.integer(error.int),
                        mark = integer(nused),
                        surv = double(nused),
                        varhaz = double(nused),
                        double(nused),
                        risksum = double(nused),
                        ntime = integer(1),
                        x,
                        double(nvar),
                        coxreg.list$var)
                ntime <- surv$ntime
                if(surv$surv[ntime] == 0 && error.int == 1)
```

```
                surv$varhaz[ntime] <- NA
        ntime <- 1:ntime
        temp <- list(time = surv$time[ntime], n.risk = surv$risksum[
                ntime], n.event = surv$mark[ntime], surv = surv$surv[
                ntime], std.err = sqrt(surv$varhaz[ntime]))
}
else {
        ttime <- NULL
        trisk <- NULL
        tn <- NULL
        tsurv <- NULL
        tvar <- NULL
        tstrat <- NULL
        for(i in 1:max(strata)) {
                who <- (strata == i)
                n <- sum(who)
                if(n == 0)
                        next
                surv <- .C("survfit",
                        as.integer(n),
                        as.integer(nvar),
                        time = stime[who],
                        sstat[who],
                        as.double(wt[who]),
                        as.integer(method),
                        as.integer(error.int),
```

```
                mark = integer(n),
                surv = double(n),
                varhaz = double(n),
                double(n),
                risksum = double(n),
                ntime = integer(1),
                x[who, ],
                double(nvar),
                coxreg.list$var)
        ntime <- surv$ntime
        if(surv$surv[ntime] == 0 && error.int == 1)
                surv$varhaz[ntime] <- NA
        ntime <- 1:ntime
        ttime <- c(ttime, surv$time[ntime])
        trisk <- c(trisk, surv$risksum[ntime])
        tn <- c(tn, surv$mark[ntime])
        tsurv <- c(tsurv, surv$surv[ntime])
        tvar <- c(tvar, surv$varhaz[ntime])
        tstrat <- c(tstrat, rep(strata.level[i], surv$ntime))
    }
    temp <- list(time = ttime, n.risk = trisk, n.event = tn, surv
         = tsurv, std.err = sqrt(tvar), strata = tstrat,
        strata.name = strata.name)
}
zval <- qnorm(1 - (1 - conf.level)/2, 0, 1)
.Options$warn <- -1     #turn off warnings since we will take log(0)
```

```
        if(conf.type == 2) {
                temp1 <- temp$surv + zval * temp$std * temp$surv
                temp2 <- temp$surv - zval * temp$std * temp$surv
                temp <- c(temp, list(upper = pmin(temp1, 1), lower = pmax(temp2,
                        0), conf.type = "plain", conf.level = conf.level))
        }
        if(conf.type == 3) {
                temp1 <- exp(log(temp$surv) + zval * temp$std)
                temp2 <- exp(log(temp$surv) - zval * temp$std)
                temp <- c(temp, list(upper = pmin(temp1, 1), lower = temp2,
                        conf.type = "log", conf.level = conf.level))
        }
        if(conf.type == 4) {
                temp1 <- exp( - exp(log( - log(temp$surv)) + (zval * temp$std)/
                        log(temp$surv)))
                temp2 <- exp( - exp(log( - log(temp$surv)) - (zval * temp$std)/
                        log(temp$surv)))
                temp <- c(temp, list(upper = temp1, lower = temp2, conf.type =
                        "log-log", conf.level = conf.level))
        }
        attr(temp, "class") <- "surv.fit"
        return(temp)
}
```

Code for PLOT.SURV.FIT:

```
function(surv, conf.int, mark.time = T, mark = 3, col = 1, lty = 1, na.strata
```

```
         = T, mark.cex = 1, log = F, yscale = 1, xlab = "Time", ylab =
        "Survival", xaxs = "e", ...)
{
        cnames <- match(c("time", "surv", "n.risk"), names(surv))
        if(any(is.na(cnames)))
                stop("`surv' must be the result of surv.fit")
        if(missing(conf.int)) {
                if(is.null(surv$strata))
                        conf.int <- T
                else conf.int <- F
        }
        strata <- surv$strata
        if(is.null(strata))
                strata <- rep(1, length(surv$time))
        strata <- as.category(strata)
        strata[is.na(strata)] <- max(strata, na.rm = T) + 1
        ngroups <- length(unique(strata))
        # set default values for missing parameters
        mark <- rep(mark, length = ngroups)
        col <- rep(col, length = ngroups)
        lty <- rep(lty, length = ngroups)
        if(is.numeric(mark.time)) mark.time <- sort(mark.time[mark.time > 0])
# for log plots we have to be tricky about the y axis scaling
#
        if(log) {
                plot(c(0, max(surv$time)), yscale * c(0.99, min(0.1, surv$surv[
```

```
                surv$surv > 0], na.rm = T)), type = "n", log = "y",
                xlab = xlab, ylab = ylab, xaxs = xaxs, ...)
    }
    else plot(c(0, max(surv$time)), yscale * c(0, 1), type = "n", xlab =
                xlab, ylab = ylab, xaxs = xaxs, ...)
    if(yscale != 1) par(usr = par("usr")/c(1, 1, yscale, yscale))   #
# put up the curves one by one
#   surv.fit has already put them into the "right" order
    i <- 0
    xend <- NULL
    yend <- NULL
    for(j in unique(strata)) {
        i <- i + 1
        who <- (strata == j)
    # next line identifies all of the 'step downs' or 'last point'
        drops <- (surv$n.event > 0 | surv$time == max(surv$time[who]))
        xx <- c(0, surv$time[who & drops])
        yy <- c(1, surv$surv[who & drops])
        lines(xx, yy, type = "s", lty = lty[i], col = col[i])
        if(is.numeric(mark.time)) {
            nn <- length(xx)
            indx <- mark.time
            for(k in seq(along = mark.time))
                indx[k] <- sum(mark.time[k] > xx)
            points(mark.time[indx < nn], yy[indx[indx < nn]], pch
                 = mark[i], col = col[i], cex = mark.cex)
```

```
            }
            else if(mark.time == T & any(surv$n.event[who] == 0))
                    points(surv$time[who & surv$n.event == 0], surv$surv[
                            who & surv$n.event == 0], pch = mark[i], col =
                            col[i], cex = mark.cex)
            xend <- c(xend, max(xx))
            yend <- c(yend, min(yy))
            if(conf.int == T && !is.null(surv$upper)) {
                    if(ngroups == 1)
                            lty[i] <- lty[i] + 1
                    yy <- c(1, surv$upper[who & drops])
                    lines(xx, yy, type = "s", lty = lty[i], col = col[i])
                    yy <- c(1, surv$lower[who & drops])
                    lines(xx, yy, type = "s", lty = lty[i], col = col[i])
            }
    }
    invisible(list(x = xend, y = yend))
}
```

A.2 Bootstrapped Kaplan-Meier Product Limit Estimator

The bootstrapped Kaplan-Meier product limit estimates are calculated using the function *km.bootstrap* and survival curves are generated using the functions *km.bootstrap.summary* and the built-in function *plot.surv.fit* (refer section A.1).

Code for KM.BOOTSTRAP

```
function(nboot, data)
```

```
{
# This function inputs the number of bootstrap iterations (nboot) and the
# data set to be analyzed (data) and outputs nboot Kaplan-Meier Survival
# Curves in the form of a list (km.results).  This output can be condensed
# by using function km.bootstrap.summary
# This function calls custom functions sampler and km.loop
        npoint <- nrow(data)
        index <- 1:npoint
        data.matrix <- data
        temp1 <- matrix(sample(index, size = length(index) * nboot, replace = T
                ), nrow = nboot)
        samples <- apply(temp1, 1, sampler, data = data.matrix)
        cat("\n", "Sampling completed.  Calculating Kaplan-Meier Survival
Curves.","\n", "\n", sep = "")
        km.results <- apply(samples, 2, km.loop)
        km.results
}
```

Subfunctions to KM.BOOTSTRAP:

Code for SAMPLER:

```
function(x, data)
{
        data.new <- matrix(c(data[x, 1], data[x, 2]), ncol = 2)
        data.new
}
```

Code for KM.LOOP:

```
function(x)
{
        row <- length(x)/2
        row.group1 <- seq(1, row, 1)
        row.group2 <- seq(row + 1, length(x), 1)
        temp1 <- x[ - row.group2]
        temp2 <- x[ - row.group1]
        data.matrix <- matrix(c(temp1, temp2), nrow = row, ncol = 2)
        data.km <- surv.fit(data.matrix[, 2], data.matrix[, 1])
        km.matrix <- print.surv.fit.dave(data.km, censored = F)
        km.matrix
}
```

Code for KM.BOOTSTRAP.SUMMARY

```
function(list.bootstrap, list.kmanal)
{
# This function inputs a bootstrapped Kaplan-Meier list (list.bootstrap) and
# a Kaplan-Meier analysis list (list.kmanl) and outputs a list (list.output)
# for each uncensored time of bootstrapped Kaplan-Meier values.
# The function calls custom functions: print.surv.fit.custom, reduce.custom,
# and list.custom
        matrix.kmanal <- print.surv.fit.custom(list.kmanal, censored = F)
        vector.kmanal <- matrix.kmanal[, 1]
        list.bootstrap.reduced <- lapply(list.bootstrap, reduce.custom)
        nboot <- length(list.bootstrap.reduced)
```

```
        output1 <- NULL

        output2 <- NULL

        for(i in 1:nboot) {

                output1 <- append(output1, list.bootstrap.reduced[[i]][, 1],

                        after = length(vector.kmanal) * (i - 1))

                output2 <- append(output2, list.bootstrap.reduced[[i]][, 2],

                        after = length(vector.kmanal) * (i - 1))

                cat("Loop #1 iteration:", i, "\n")

        }

        boot.matrix <- matrix(c(output1, output2), ncol = 2)

        boot.matrix.sort <- boot.matrix[order(boot.matrix[, 1]), 1:2]

        list.temp1 <- split(vector.kmanal, vector.kmanal)

        data.point <- nrow(boot.matrix.sort)

        list.output <- lapply(list.temp1, list.custom, vector.search =

                vector.kmanal, matrix.search = boot.matrix.sort, npoint =

                data.point)
```

Subfunctions to KM.BOOTSTRAP.SUMMARY

Code for PRINT.SURV.FIT.CUSTOM:

```
function(fit, times = NULL, censored = F, digits = NULL)

{

        cnames <- match(c("time", "n.risk", "n.event", "surv", "std.err"),

                names(fit))

        if(any(is.na(cnames)))

                stop("Argument must be the result of 'surv.fit'")
```

```
        if(!missing(digits)) {
                if((length(digits) != 1 || digits < 1) || digits > 20)
                        stop("Bad value for digits argument")
                d <- options(digits = digits)
                on.exit(options(d))
        }
        std.err <- fit$std.err * fit$surv
        if(is.null(times)) {
                if(!is.null(fit$lower))
                        mat <- cbind(fit$time, fit$n.risk, fit$n.event, fit$
                                surv, std.err, fit$lower, fit$upper)
                else mat <- cbind(fit$time, fit$n.risk, fit$n.event, fit$surv,
                                std.err)
                if(!censored)
                        mat <- mat[fit$n.event > 0,  , drop = F]
                if(!is.null(fit$strata)) {
                        if(censored)
                                strata <- fit$strata
                        else strata <- fit$strata[fit$n.event > 0]
                }
        }
        else {
#this case is much harder
                if(any(times < 0)) stop("Invalid time point requested")
                if(length(times) > 1)
                        if(any(diff(times) < 0))
```

```
                        stop("Times must be in increasing order")
        n <- length(fit$surv)
        if(is.null(fit$strata))
                stemp <- rep(1, n)
        else stemp <- as.category(fit$strata)
        stemp[is.na(stemp)] <- 0
#let missing endure as a valid category
        nn <- length(unique(stemp))
        temp <- .C("survindex",
                as.integer(n),
                as.double(fit$time),
                as.integer(stemp),
                as.integer(length(times)),
                as.double(times),
                as.integer(nn),
                indx = integer(nn * length(times)),
                indx2 = integer(nn * length(times)))
        keep <- temp$indx >= 0
        indx <- temp$indx[keep]
        ones <- (temp$indx2 == 1)[keep]
        ties <- (temp$indx2 == 2)[keep]
        times <- rep(times, nn)[keep]
        n.risk <- fit$n.risk[indx + 1 - (ties + ones)]
        surv <- fit$surv[indx]
        surv[ones] <- 1
        std.err <- std.err[indx]
```

```
            std.err[ones] <- 0
            fit$n.event[fit$time > max(times)] <- 0
            n.event <- (cumsum(c(0, fit$n.event)))[ifelse(ones, indx,
 indx + 1)]
            n.event <- diff(c(0, n.event))
            if(is.null(fit$lower))
                    mat <- cbind(times, n.risk, n.event, surv, std.err)
            else {
                    lower <- fit$lower[indx]
                    lower[ones] <- 1
                    upper <- fit$upper[indx]
                    upper[ones] <- 1
                    mat <- cbind(times, n.risk, n.event, surv, std.err,
                            lower, upper)
            }
            if(!is.null(fit$strata))
                    strata <- fit$strata[indx]
    }
    if(is.null(fit$lower))
            dimnames(mat) <- list(NULL, c("time", "n.risk", "n.event",
                    "survival", "std.dev"))
    else {
            dimnames(mat) <- list(NULL, c("time", "n.risk", "n.event",
                    "survival", "std.dev", paste("lower ",
fit$conf.level * 100, "% CI", sep = ""),
paste("upper ", fit$conf.level * 100, "% CI",
```

```
sep = "")))
        }
        mat
}
```

Code for REDUCE.CUSTOM:

```
function(list.bootstrap)
{
        matrix.data <- matrix(c(list.bootstrap[, 1], list.bootstrap[, 4]), ncol
                = 2)
        matrix.data
}
```

Code for LIST.CUSTOM:

```
function(time, vector.search, matrix.search, npoint)
{
        vector <- NULL
        counter <- 0
        for(i in 1:npoint) {
                if(matrix.search[i, 1] == time[1]) {
                        vector <- append(vector, matrix.search[i, 2], after =
                                counter)
                        counter <- counter + 1
                }
        }
        cat("Loop #2 iteration for time", time[1], "\n")
        vector
```

```
}
```

A.3 Nelson's Cumulative Hazard Plotting Technique

Nelson's cumulative hazard plotting technique is implemented using the custom fuction *hazard.tech*.

Code for HAZARD.TECH:

```
function(data)
{
#  Function inputs a data matrix (data) containing failure data to be analyzed
#  Column 1 of the matrix contains the censoring information about the data
#  (0 indicates a right censored observation and 1 indicates a uncensored
#  [failure] observation). Column 2 of the matrix denotes observation times.
#  Function outputs the non-parametric hazard rate and cumulative hazard rate
#  estimators for the uncensored observations
        temp1 <- matrix(c(data[, 2], data[, 1]), ncol = 2)
        temp2 <- temp1[order(temp1[, 1]), 1:2]
        temp3 <- cbind(temp2, seq(nrow(temp2), 1, -1))
        vector.time <- NULL
        vector.hazard.rate <- NULL
        vector.cum.hazard <- NULL
        cum.hazard <- 0
        counter <- 0
        for(i in 1:nrow(temp3)) {
                if(temp3[i, 2] == 1) {
                        hazard.rate <- 1/temp3[i, 3]
```

```
                cum.hazard <- cum.hazard + hazard.rate
                vector.time <- append(vector.time, temp3[i, 1], after
                         = counter)
                vector.hazard.rate <- append(vector.hazard.rate,
                        hazard.rate, after = counter)
                vector.cum.hazard <- append(vector.cum.hazard,
                        cum.hazard, after = counter)
                counter <- counter + 1
        }
    }
    output <- matrix(c(vector.time, vector.hazard.rate, vector.cum.hazard)
, ncol = 3)
    output
}
```

A.4 Lawson's Non-Parametric Bayesian Estimator Using a Gibbs Sampling Algorithm

Lawson's non-parametric bayesian estimator using a gibbs sampling algorithm is coded using FORTRAN77.

```
c program lawsongibbs.f


parameter(maxn=1000, maxtry=100,iter=50,writiter=10)
integer maxtry,m,c,number,index,mstar,cstar,nstar,run,zelement,ndisc
integer iter, counter, counter2
real y(maxn),z(maxn),r(maxn),tempv1(maxn),ystar(maxn),zstar(maxn)
real alpha,prob,a,b,sumystar,sumzstar,sumxstar, pdraw, discrete(maxn)
```

```
real tempr,oldz,lambda
character*100 fail,right

c  Prompt user for:
c m - number of observed failures
c c - number of right censored observations
c fail - name of the file containing the observed failures
c right - name of the file containing the right censored observations

print*,"Enter the number of observed failures -->"
read*,m
print*,"Enter the number of right censored observations -->"
read*,c
n = m + c
print*,"Enter the name of the input file that contains the observed
failures -->"
read(*,500) fail
print*,"Enter the name of the input file that contains right censored
 observations -->"
read(*,500) right


c  Create vectors:
c y - vector of observed failures
c z - vector of calculated failures for right censored observations
c r - vector of right censored observations
```

```
c  CAUTION: make sure format statements in 505 and 506 agree with inputed data.
open(1,file=fail)
read(1,505) (y(i),i=1,m)
open(2,file=right)
read(2,506) (z(i),r(i),i=1,c)


c  Prompt user
c alpha - constant used to determine frequency of picking from the
parametric distribution
c a,b - priors for parametric distribution


print*,"Enter the desired value of alpha --->"
read *,alpha
prob = alpha / (alpha + (n-1))
print*,"Enter the desired priors a & b --->"
read *,a,b
print*,"Enter the number of iterations --->"
read *,number


c - cstar is the number of unique elements in z


mstar = m
call minimize(ystar,mstar,y,m)


call remove(tempv1,1,z,c)
cstar = c
```

```
call minimize(zstar,cstar,tempv1,c-1)

sumystar = 0
do 5 index = 1,mstar
sumystar = ystar(index) + sumystar
 5 continue

sumzstar =0
do 10 index = 1,cstar
sumzstar = zstar(index) + sumzstar
 10 continue

sumxstar = sumystar + sumzstar
nstar = mstar + cstar

open(1,file="gibbs.output")

c       Beginning Loop for iterations

print 500, "Completed iteration: "
do 20 run = 1,number

lambda = a/b
counter2 = 0
c Beginning Loop for updating elements of vector z
```

```
do 15 zelement = 1,c
pdraw = uni()
counter = 0
oldz = z(zelement)
if (pdraw .le. prob) then
30 counter = counter +1
z(zelement) = rexp()/lambda
if ((z(zelement) .lt. r(zelement)).and.(counter
     + .lt. maxtry)) go to 30
else
discrete(1) = 0
do 35 index = 1,cstar
tempr = real(index)
discrete(index+1) = tempr/cstar
35 continue
40 counter = counter +1
  ndisc = nonpar(discrete,cstar+1)
z(zelement) = zstar(ndisc)
if ((z(zelement) .lt. r(zelement)) .and.
     + (counter.lt. maxtry)) go to 40
endif
if (counter.ge.maxtry) then
z(element) = oldz
endif
call remove(tempv1,1,z,c)
cstar = c
```

```
call minimize(zstar,cstar,tempv1,c-1)

sumzstar =0

do 45 index = 1,cstar

   sumzstar = zstar(index) + sumzstar

45 continue

sumxstar = sumystar + sumzstar

nstar = mstar + cstar

lambda = rgamma(a+nstar)/(b+sumxstar)

15 continue

if (mod(run,iter).eq.0) then

     print *,run,"lambda =",lambda

endif

if (mod(run,writiter).eq.0) then

write(1,520),(y(index),index=1,m)

write(1,520),(z(index),index=1,c)

50 continue

endif

20 continue

close (1)


500  format(a)

505 format(f10.1)

506 format(2f7.1)

510     format(i5)

520 format(f10.2)

530 format(f10.9)
```

```
stop
end

************************************************************

subroutine remove(redvec,val,vector,size)

c   This subrountine removes the vector element specified in "val"

integer size, val
real redvec(1000),vector(size)

c   redvec - the remaining vector elements when the element specified by "val"
c            is removed
c    val - the vector element to be removed
c   vector - complete vector
c   size - length of "vector"

if (val .eq. 1) then
do 5 index = 2,size
redvec(index-1) = vector(index)
 5 continue
else
do 10 index = 1,(val-1)
redvec(index) = vector(index)
```

```
 10 continue
do 15 index = (val+1),size
redvec(index-1) = vector(index)
 15 continue
end if
return
end

**************************************************************

subroutine minimize(minvec,minsize,vector,size)

c   This subrountine reduces a vector to its unique elements

integer minsize,size,index,index1,index2
real minvec(minsize),vector(size),temp1(1000),temp2(1000)

c   minvec - the vector of unique elements
c   minsize - length of "minvec"
c   vector - complete vector
c   size - length of "vector"

do 5 index = 1,size
temp1(index) = vector(index)
 5 continue
call sort(temp1,size)
```

```
temp2(1) = temp1(1)
index1 = 1
index2 = 2
 10 if (temp2(index1) .lt. temp1(index2)) then
temp2(index1 +1) = temp1(index2)
index1 = index1 +1
endif
index2 = index2 +1
if ((index2 - 1) .lt. size) go to 10
minsize = index1
do 15 index = 1,minsize
minvec(index) = temp2(index)
 15 continue
return
end

*************************************************************************

subroutine sort (vector,size)

c  index1,index2 - loop indices
c  imin - current position of the minimum element
c  mover - The minimum value in the postion imin

integer size,index1, index2, imin
real vector(1:size), mover
```

```
imin = minpos(vector,size)
mover = vector(imin)
vector(imin) = vector(1)
vector(1) = mover
do 20 index1 = 3,size
mover = vector(index1)
index2 = index1
 10 if (vector(index2-1) .gt. mover) then
vector(index2) = vector(index2-1)
index2 = index2 - 1
go to 10
end if
vector(index2) = mover
 20 continue
end

*************************************************************************
function minpos (vector,size)

c  Finds the subscript of vector element having the lowest value

c  Index - Loop index
c  minval - the currently known minimum value

integer size,index, minval
real vector(1:size)
```

```
minval = vector(1)
minpos = 1
do 10 index = 2, size
if (vector(index) .lt. minval) then
minval = vector(index)
minpos = index
end if
 10 continue
end

***********************************************************************
function nonpar (vector,size)

integer size, nonpar
real vector(size), draw

draw = uni()
nonpar = 0
if (draw .eq. 0) then
    nonpar = 1
else
          do 10 index = 1,size-1
          if ((draw .gt. vector(index)).and.(draw .le. vector(index+1)))
     +  then
nonpar = index
```

```
   endif
 10   continue
 endif
 return
 end

***********************************************************************

c     The following are random generator subroutines and functions
c     needed to carry out the simulation.

          function rgamma(a)
c
c Returns a Gamma variate using the SQUEEZE method
c G. Marsaglia,  Comp. & Maths with Applns. Vol 3. pp 321-325, 1977.
c
c Note: The argument should be greater than 1/3.
c
      data b/-1./
      if (a.eq.1.0) then
        rgamma=rexp()
        return
      endif
      if (b.eq.a) go to 1
      b=a
      s=.3333333/sqrt(a)
```

```
      z0=1.-1.732051*s
      cc=a*z0**3-.5*(s-1.732051)**2
      cl=3*a-1.
      cs=1.-s*s
    1 x=rnor()
      z=s*x+cs
      if (z.le.0) go to 1
      rgamma=a*z**3
      e=rexp()
      cd=e+.5*x**2-rgamma+cc
      t=1.-z0/z
      if(cd+cl*t*(1.+t*(.5+.3333333*t)).gt.0.) return
      if(cd+cl*alog(z/z0).lt.0.) go to 1
      return
      end

***********************************************************************
      function rnor()
c
c Returns a standard Normal variate using the Ziggurat method.
c G. Marsaglia & Wai Tsang, Siam J. Sci. Stat. Comput. Vol 5
c June 1984, pp 349-359.
c
c April 5, 1990. Made a correction to the tail part so that
c Log of a zero UNI is avoided.
c March 5, 1991. Made a correction to c2.       B. Narasimhan
```

```
c
      real v(0:256)
      DATA AA,B,C/25.74023263217, .3194187376868, 25.96701302767/
      DATA RMAX/5.960464478e-8/
      DATA C1,C2,PC,XN/.9674937559244, 1.226780395498,
     +                 .00489575835,  3.289869847629/
c
      data (v(j), j=0, 29)/
     + .24698083002457, .30686964662795, .35153978938895,
     + .38807372507154, .41944111113146, .44719524398379,
     + .47226002099856, .49523381630958, .51652849544015,
     + .53644082893859, .55519243431656, .57295362330211,
     + .58985839761905, .60601428557041, .62150902494419,
     + .63641523999656, .65079379879381, .66469627686500,
     + .67816680018702, .69124344747922, .70395933341549,
     + .71634345674087, .72842137244342, .74021573037902,
     + .75174671122491, .76303238257455, .77408899225176,
     + .78493121178444, .79557233995228, .80602447408300/
c
      data (v(j), j=30, 59)/
     + .81629865509377, .82640499100558, .83635276268797,
     + .84615051484361, .85580613466031, .86532692010181,
     + .87471963944859, .88399058341281, .89314561092262,
     + .90219018948527, .91112943088938, .91996812288381,
     + .92871075737076, .93736155556822, .94592449052850,
     + .95440330734290, .96280154131524, .97112253434701,
```

```
     + .97936944974390, .98754528562459, .99565288708942,
     +1.00369495728548,1.01167406748786,1.01959266630145,
     +1.02745308807506,1.03525756060863,1.04300821222465,
     +1.05070707826663,1.05835610708045,1.06595716552808/
c
      data (v(j), j=60, 89)/
     +1.07351204407750,1.08102246150823,1.08849006926751,
     +1.09591645550830,1.10330314883748,1.11065162179932,
     +1.11796329411693,1.12523953571236,1.13248166952351,
     +1.13969097413490,1.14686868623717,1.15401600292913,
     +1.16113408387481,1.16822405332690,1.17528700202672,
     +1.18232398899035,1.18933604318933,1.19632416513391,
     +1.20328932836597,1.21023248086826,1.21715454639590,
     +1.22405642573584,1.23093899789935,1.23780312125216,
     +1.24464963458674,1.25147935814061,1.25829309456444,
     +1.26509162984343,1.27187573417500,1.27864616280591/
c
      data (v(j), j=90, 119)/
     +1.28540365683140,1.29214894395899,1.29888273923931,
     +1.30560574576607,1.31231865534737,1.31902214915020,
     +1.32571689831988,1.33240356457626,1.33908280078811,
     +1.34575525152740,1.35242155360456,1.35908233658634,
     +1.36573822329735,1.37238983030648,1.37903776839932,
     +1.38568264303767,1.39232505480705,1.39896559985334,
     +1.40560487030916,1.41224345471127,1.41888193840939,
     +1.42552090396760,1.43216093155888,1.43880259935355,
```

```
     +1.44544648390252,1.45209316051569,1.45874320363660,
     +1.46539718721367,1.47205568506879,1.47871927126394/
c
      data (v(j), j=120, 149)/
     +1.48538852046644,1.49206400831339,1.49874631177596,
     +1.50543600952421,1.51213368229290,1.51883991324899,
     +1.52555528836149,1.53228039677414,1.53901583118162,
     +1.54576218820999,1.55252006880182,1.55929007860681,
     +1.56607282837854,1.57286893437799,1.57967901878452,
     +1.58650371011513,1.59334364365258,1.60019946188339,
     +1.60707181494618,1.61396136109153,1.62086876715397,
     +1.62779470903712,1.63473987221291,1.64170495223585,
     +1.64869065527338,1.65569769865342,1.66272681143024,
     +1.66977873496983,1.67685422355612,1.68395404501935/
c
      data (v(j), j=150, 179)/
     +1.69107898138789,1.69822982956533,1.70540740203402,
     +1.71261252758712,1.71984605209074,1.72710883927814,
     +1.73440177157803,1.74172575097914,1.74908169993340,
     +1.75647056230006,1.76389330433365,1.77135091571826,
     +1.77884441065150,1.78637482898114,1.79394323739797,
     +1.80155073068867,1.80919843305257,1.81688749948677,
     +1.82461911724403,1.83239450736871,1.84021492631583,
     +1.84808166765936,1.85599606389581,1.86395948835000,
     +1.87197335719041,1.88003913156195,1.88815831984492,
     +1.89633248004955,1.90456322235616,1.91285221181234/
```

```
c
      data (v(j), j=180, 209)/
     +1.92120117119893,1.92961188407826,1.93808619803888,
     +1.94662602815264,1.95523336066136,1.96391025691183,
     +1.97265885756012,1.98148138706769,1.99038015851462,
     +1.99935757875730,2.00841615396118,2.01755849554212,
     +2.02678732655353,2.03610548856055,2.04551594904689,
     +2.05502180940521,2.06462631356762,2.07433285733931,
     +2.08414499850583,2.09406646779286,2.10410118076694,
     +2.11425325077645,2.12452700304473,2.13492699004145,
     +2.14545800827495,2.15612511666698,2.16693365669349,
     +2.17788927450032,2.18899794523217,2.20026599984763/
c
      data (v(j), j=210, 239)/
     +2.21170015473319,2.22330754447619,2.23509575821221,
     +2.24707288002746,2.25924753397443,2.27162893435095,
     +2.28422694200254,2.29705212753971,2.31011584252007,
     +2.32343029983670,2.33700866478640,2.35086515857565,
     +2.36501517636968,2.37947542241971,2.39426406533604,
     +2.40940091723926,2.42490764135740,2.44080799369296,
     +2.45712810572956,2.47389681687590,2.49114606758128,
     +2.50891136697751,2.52723235275195,2.54615346608274,
     +2.56572477136742,2.58600295987340,2.60705258940068,
     +2.62894763017268,2.65177341290207,2.67562911211061/
c
      data (v(j), j=240, 256)/
```

```
     +2.70063095234582,2.72691640673304,2.75464978267448,
     +2.78402978650249,2.81529997720240,2.84876355022267,
     +2.88480481063120,2.92392135126341,2.96677409038864,
     +3.01426863112874,3.06769501549787,3.12898502985483,
     +3.20123093035120,3.28986984762935,3.28986984762935,
     +3.28986984762935,3.28986984762935/
c
      i=ivni()
      j=iand(i,255)
      rnor=i*rmax*v(j+1)
      if (abs(rnor).le.v(j)) return
      X = (ABS(rnor)-V(J))/(V(J+1)-V(J))
      Y=UNI()
      S=X+Y
      IF (S .GT. C2) GO TO 11
      IF (S .LE. C1) RETURN
      IF (Y .GT. C-AA*EXP(-.5*(B-B*X)**2)) GO TO 11
      IF (EXP(-.5*V(J+1)**2)+Y*PC/V(J+1) .LE. EXP(-.5*rnor**2))
     +RETURN
c      ----------------TAIL PART---------------------

22    x=uni()
      if (x.eq.0.) goto 22
      x=.3039633925703*alog(x)
23    y=uni()
      if (y.eq.0.) goto 23
```

```
      if(-2.*alog(y).le.x*x) goto 22
      rnor = SIGN(XN-X,rnor)
      RETURN
11    rnor = SIGN(B-B*X,rnor)
      RETURN
      END

*********************************************************************
      function rexp()
c
c Returns a standard Exponential variate using the Ziggurat method.
c G. Marsaglia & Wai Tsang, Siam J. Sci. Stat. Comput. Vol 5
c June 1984, pp 349-359.
c
c April 5, 1990. Made a correction to the tail part so that
c Log of a zero UNI is avoided.
c March 5, 1991, made a correction to c2.   B. Narasimhan
c
      real v(0:256)
      DATA A,B,C/8.251197925281462, 0.129031492286153,
     +                               8.269609187493524/
      data rmax/5.960464478e-8/
      DATA C1,C2,P,XN/0.913928667944066,1.033948296676074,
     +                .00390625,      7.569274694148/
c
      data (v(j), j=0, 29)/
```

```
     + .126655859648059, .155569621307508, .181093717722647,
     + .204278845879366, .225733963189410, .245847546956213,
     + .264884646355367, .283035319979449, .300441176903090,
     + .317210988594029, .333430399046011, .349168242727458,
     + .364480811271385, .379414826381335, .394009567169737,
     + .408298427790513, .422310080983936, .436069362664599,
     + .449597954997061, .462914921252629, .476037129872182,
     + .488979594498425, .501755749432899, .514377674872274,
     + .526856282659199, .539201470676226, .551422252107873,
     + .563526864388270, .575522861598826, .587417193283927/
c
      data (v(j), j=30, 59)/
     + .599216272044407, .610926031799538, .622551978243536,
     + .634099232736614, .645572570644684, .656976454962030,
     + .668315065907273, .679592327066830, .690811928565931,
     + .701977347670446, .713091867159808, .724158591759403,
     + .735180462877832, .746160271858668, .757100671926444,
     + .768004188981551, .778873231377573, .789710098796722,
     + .800516990323877, .811296011806816, .822049182579159,
     + .832778441613126, .843485653161020, .854172611937377,
     + .864841047887595, .875492630583650, .886128973282865,
     + .896751636681758, .907362132393472, .917961926174235/
c
      data (v(j), j=60, 89)/
     + .928552440921646, .939135059465172, .949711127167206,
     + .960281954351162, .970848818571479, .981412966738940,
```

```
+ .991975617113456,1.002537961175289,1.013101165384700,
+1.023666372839080,1.034234704835818,1.044807262348423,
+1.055385127422776,1.065969364499780,1.076561021670175,
+1.087161131866780,1.097770713999020,1.108390774034195,
+1.119022306029592,1.129666293119249,1.140323708458863,
+1.150995516132092,1.161682672021260,1.172386124645272,
+1.183106815967315,1.193845682174803,1.204603654433794,
+1.215381659620038,1.226180621028596,1.237001459063937/
c
 data (v(j), j=90, 119)/
+1.247845091912237,1.258712436197555,1.269604407623440,
+1.280521921601448,1.291465893867999,1.302437241090886,
+1.313436881466735,1.324465735310640,1.335524725639139,
+1.346614778747679,1.357736824783652,1.368891798316067,
+1.380080638902888,1.391304291657037,1.402563707812060,
+1.413859845288389,1.425193669261187,1.436566152730679,
+1.447978277095918,1.459431032732909,1.470925419577996,
+1.482462447717458,1.494043137984220,1.505668522562623,
+1.517339645602190,1.529057563841350,1.540823347242078,
+1.552638079636446,1.564502859386106,1.576418800055707/
c
 data (v(j), j=120, 149)/
+1.588387031101344,1.600408698575105,1.612484965846851,
+1.624617014344384,1.636806044313222,1.649053275597203,
+1.661359948441241,1.673727324317561,1.686156686776822,
+1.698649342325602,1.711206621331769,1.723829878959332,
```

```
     +1.736520496134471,1.749279880544484,1.762109467671511,
     +1.775010721862970,1.787985137440760,1.801034239851363,
     +1.814159586859135,1.827362769785157,1.840645414794192,
     +1.854009184232407,1.867455778018688,1.880986935092532,
     +1.894604434921695,1.908310099072946,1.922105792849485,
     +1.935993426998818,1.949974959495098,1.964052397400219/
c
      data (v(j), j=150, 179)/
     +1.978227798808186,1.992503274877628,2.006880991957598,
     +2.021363173812162,2.035952103949651,2.050650128062839,
     +2.065459656586748,2.080383167381237,2.095423208546051,
     +2.110582401376519,2.125863443468717,2.141269111983517,
     +2.156802267079656,2.172465855526694,2.188262914509569,
     +2.204196575637314,2.220270069169488,2.236486728474926,
     +2.252849994738532,2.269363421933134,2.286030682074740,
     +2.302855570781062,2.319842013154793,2.336994070014906,
     +2.354315944501240,2.371811989079741,2.389486712978150,
     +2.407344790084496,2.425391067343645,2.443630573690306/
c
      data (v(j), j=180, 209)/
     +2.462068529560411,2.480710357026633,2.499561690608087,
     +2.518628388809013,2.537916546446482,2.557432507833039,
     +2.577182880886687,2.597174552247873,2.617414703491264,
     +2.637910828529106,2.658670752313152,2.679702650953474,
     +2.701015073385286,2.722616964729252,2.744517691506983,
     +2.766727068891732,2.789255390195018,2.812113458813384,
```

```
+2.835312622886206,2.858864812945841,2.882782582876061,
+2.907079154534361,2.931768466439166,2.956865226975109,
+2.982384972629692,3.008344131843985,3.034760095140333,
+3.061651292283300,3.089037277338545,3.116938822621064/
c
 data (v(j), j=210, 239)/
+3.145378022672564,3.174378409582095,3.203965081169483,
+3.234164843794194,3.265006371840749,3.296520386275642,
+3.328739855082013,3.361700218872422,3.395439645576226,
+3.429999318820512,3.465423765503387,3.501761229135142,
+3.539064096847590,3.577389389610734,3.616799327235384,
+3.657361982293938,3.699152040309946,3.742251687651506,
+3.786751653785457,3.832752441277892,3.880365785669281,
+3.929716398814248,3.980944064446654,4.034206175017578,
+4.089680826283840,4.147570623658259,4.208107406359190,
+4.271558168532011,4.338232560910351,4.408492508122202/
c
 data (v(j), j=240, 256)/
+4.482764700811059,4.561557059876923,4.645480792126252,
+4.735280483126629,4.831876019703573,4.936422401033893,
+5.050397456688548,5.175734711481234,5.315032491954608,
+5.471898603959591,5.651551824524737,5.861950186930114,
+6.116117818708264,6.437612978975440,6.876127513588117,
+7.569274694148062,7.569274694148062/
c
 i=iuni()
```

```
      j=iand(i,255)
      rexp=i*rmax*v(j+1)
      if (rexp.le.v(j)) return
      X = (rexp-V(J))/(V(J+1)-V(J))
      Y=UNI()
      S=X+Y
      IF (S .GT. C2) GO TO 11
      IF (S .LE. C1) RETURN
      IF (Y .GT. C-A*EXP(B*X-B)) GO TO 11
      IF (EXP(-V(J+1))+Y*P/V(J+1) .LE. EXP(-rexp))
     +RETURN
c      ----------------TAIL PART----------------------
2     x = uni()
      if (x.eq.0.) goto 2
      rexp=XN-ALOG(x)
      return
11    rexp=B-B*X
      RETURN
      END

************************************************************************
       function uni()
c The uni function sub-program combines, with subtraction mod 1,
c an f(97,33,-mod 1) generator with the element c in the arithmetic
c sequence generated by c=c-cd mod(16777213./16777216.), period 2**24-3.
c period of combined generator is (2**97-1)(2**24-3)2**23, about 2**144.
```

```
      external unidata
      integer u(97),c,vnisign,i,j
      common /unidat/ u,c,vnisign,i,j
      iu=iand(u(i)-u(j),16777215)
      u(i)=iu
      i=i-1
      if(i.eq.0) i=97
      j=j-1
      if(j.eq.0) j=97
      c=c-7654321
      if(c.lt.0) c=c+16777213
      uni=iand(iu-c,16777215)/16777216.0
      vnisign = iand(iu,32)
      return
      end

*********************************************************************
      function iuni()
c The iuni function sub-program combines, with subtraction mod 2**24,
c an f(97,33,-mod 2**24) generator with the element c in the arithmetic
c sequence generated by c=c-cd mod(16777213), period 2**24-3.
c period of combined generator is (2**97-1)(2**24-3)2**23, about 2**144.
      external unidata
      integer u(97),c,vnisign,i,j
      common /unidat/ u,c,vnisign,i,j
      iuni=iand(u(i)-u(j),16777215)
```

```
      u(i)=iuni
      vnisign = iand(iuni,32)
      i=i-1
      if(i.eq.0) i=97
      j=j-1
      if(j.eq.0) j=97
      c=c-7654321
      if(c.lt.0) c=c+16777213
      iuni=iand(iuni-c,16777215)
      return
      end

*******************************************************************
      function ivni()
c The ivni function sub-program combines, with subtraction mod 2**24,
c an f(97,33,-mod 2**24) generator with the element c in the arithmetic
c sequence generated by c=c-cd mod(16777213), period 2**24-3.
c period of combined generator is (2**97-1)(2**24-3)2**23, about 2**144.
      external unidata
      integer u(97),c,vnisign,i,j
      common /unidat/ u,c,vnisign,i,j
      ivni=iand(u(i)-u(j),16777215)
      u(i)=ivni
      vnisign = iand(ivni,32)
      i=i-1
      if(i.eq.0) i=97
```

```
      j=j-1
      if(j.eq.0) j=97
      c=c-7654321
      if(c.lt.0) c=c+16777213
      ivni=iand(ivni-c,16777215)
      if(vnisign .ne. 0) ivni = -ivni
      return
      end

*********************************************************************
      block data unidata
c
c Initialized values in COMMON block for UNI, VNI, IUNI, IVNI
c and RSTART.
c
      integer u(97),c,vnisign,ip,jp
      common /unidat/ u,c,vnisign,ip,jp
      data c,vnisign,ip,jp /362436,0,97,33/
      data (u(j), j=1,97)/
     +13697435, 3833429,12353926, 2287754, 3468638, 1232959, 8059805,
     +10745739, 4236676, 2095136, 1349346, 3672867,14563641,15473517,
     + 9897259, 2207061,  929657, 8109095, 5246947, 1066111, 8460236,
     +13162386,  501474,10402355,  352505, 2104170,12045925, 4350943,
     +13996856, 9897761, 6626452,15057436, 3168599,14038489, 8550848,
     + 5242835,13296102,11969002,   95246, 5917978, 8555838,13557738,
     + 1526088,11197237,15721125,14247931,  897046,15537441,16645456,
```

```
+16279884, 1289925,14032128,10641039, 9961793, 2737638, 5073398,
+ 5231619, 2007688,15753584,12368695,12926325,10522018, 8692194,
+ 8531802,14755384,  276334, 9157821,  989353, 6093627,15866666,
+ 9532882, 3434034,  710155,  672726,12734991,13809842, 4832132,
+ 9753458,11325486,12137466, 3617374, 4913050, 9978642,12740205,
+15754026, 4928136, 8545553,12893795, 8164497,12420478, 8192378,
+ 2028808, 1183983, 3474722,15616920,16298670,14606645/
 end
```

A.5 Log Rank (Mantel) Test and Peto-Peto Modification to the Wilcoxon Test

The log rank test and Peto-Peto modification to the Wilcoxon test were conducted using the built-in function *surv.diff*. Setting $rho = 0$ performs the log rank test. Setting $rho = 1$ performs the Peto-Peto modification to the Wilcoxon test.

```
function(time, status, group, rho = 0)
{
        group <- unclass(group)
        status <- unclass(status)
        n <- length(group)
        if(length(status) != n || length(time) != n)
                stop("No. of observations in time, status, and group must match"
                        )
        if(length(rho) != 1 || !(is.numeric(rho))) stop(
                        "Invalid value for 'rho'")      #
# find observations with missing values
#   and check for legal time and status values
```

```
nomiss <- (is.finite(time) & is.finite(status) & is.finite(group))
n <- sum(nomiss)
if(any(time[nomiss] < 0))
        stop("Time values must be >=0")
zz <- status[nomiss]
if(any(zz > 1))
        zz <- zz - 1
if(all(zz == 1))
        warning("Only one status given, taken as uncensored")
if(any(zz != 0 & zz != 1))
        stop("Invalid status value")
ttime <- time[nomiss]
tstat <- zz
tgrp <- as.category(group[nomiss])
ngroup <- length(levels(tgrp))
if(ngroup < 2)
        stop("There is only 1 group")
ord <- order(ttime)
xx <- .C("survdiff",
        as.integer(n),
        as.integer(ngroup),
        as.double(rho),
        as.double(ttime[ord]),
        as.integer(tstat[ord]),
        as.integer(tgrp[ord]),
        observed = double(ngroup),
```

```
                expected = double(ngroup),
                var.e = double((ngroup - 1) * (ngroup - 1)),
                double(ngroup),
                double(n))
        temp <- table(tgrp, tstat)
        if(ncol(temp) == 1)
                temp <- cbind(0, temp)
        temp2 <- (xx$observed - xx$expected)[-1]
        chi <- sum(solve(matrix(xx$var.e, ncol = ngroup - 1), temp2) * temp2)
        retlist <- list(n = temp[, 1] + temp[, 2], obs = xx$observed, exp = xx$
                expected, chisq = chi)
        attr(retlist, "class") <- "surv.diff"
        retlist
}
```

A.6 Cox Proportional Hazards Model Coefficient Tests

The Cox Proportional Hazards Model coefficient tests were performed using the built-in function *coxreg*.

```
function(time, status, x, strata = rep(1, length(time)), wt = rep(1, length(
        time)), resid = "none", init = rep(0, ncol(x)), iter.max = 10, eps =
        0.0001, table.n = T, inf.ratio = 2000)
{
        status <- unclass(status)
        strata <- unclass(strata)
        resid.int <- charmatch(resid, c("none", "martingale", "deviance",
```

```
                "score", "schoenfeld"))
        if(is.na(resid.int))
                stop("Invalid residual type specified")
        if(resid.int == 0)
                stop("Ambiguous residual type")
        x <- as.matrix(x)
        n <- nrow(x)
        nvar <- ncol(x)
        if(length(status) != n || length(time) != n)
                stop("No. of observations in time, status, and x must match")
        if(length(strata) != n)
                stop("Strata vector is the wrong length")
        if(length(wt) != n)
                stop("Wt vector is the wrong length")
        if(length(init) != nvar) stop(
                        "Vector of initial coefficients is the wrong length")   #
# find observations with missing values
#   and check for legal time and status values
#       nomiss <- !(is.na(time) | is.na(status) | is.na(x %*% rep(1, nvar)) |
#               is.na(strata) | is.na(wt))
        nomiss <- (is.finite(time) & is.finite(status) & is.finite(wt) &
                is.finite(strata) & is.finite(x %*% rep(1, nvar)))
        nused <- sum(nomiss)
        if(any(time[nomiss] < 0))
                stop("Time values must be >= 0")
        zz <- status[nomiss]
```

```
        if(any(zz > 1)) {

                zz <- zz - 1

                status[nomiss] <- zz

        }

        else {

                if(all(zz == 1))

                        warning("only one status given, taken as uncensored")

        }

        if(any(zz != 0 & zz != 1)) stop("Invalid status value") #

# Make the outcomes table

#

        temp <- category(ifelse(nomiss, status, 2), levels = 0:2, labels = c(

                "Alive", "Dead", "Deleted"))

        if(missing(strata)) n.table <- table(temp) else n.table <- table(strata,

                        temp)

        # Sort the data (or rather, get a list of sorted indices)

        sorted <- ((1:n)[nomiss])[order(strata[nomiss], time[nomiss])]

        # create the "newstrat" vector, which is 1 at the end of each strata

        newstrat <- as.integer(c(1 * (diff(strata[sorted]) != 0), 1))

        # Subtract the mean from all covars, as this makes the regression much

#  more stable.

# I originally used "apply" and "sweep"-- boy are they slow!

        xx <- as.matrix(x[sorted,  ])

        for(i in 1:nvar)

                xx[, i] <- xx[, i] - mean(xx[, i])

        temp <- apply(abs(xx), 2, mean) #mean abs deviation from the mean
```

```
maxbeta <- log(inf.ratio)/temp
storage.mode(xx) <- "double"
stime <- as.double(time[sorted])
sstat <- as.integer(status[sorted])
coxfit <- .C("coxfit",
        iter = as.integer(iter.max),
        as.integer(nused),
        as.integer(nvar),
        stime,
        sstat,
        xx,
        as.double(wt[sorted]),
        newstrat,
        coef = as.double(init),
        double(nvar),
        as.double(maxbeta),
        imat = double(nvar * nvar),
        loglik = double(2),
        flag = integer(1),
        mark = integer(nused),
        double(nvar * (nvar + 2)),
        as.double(eps),
        sctest = double(1))
if(coxfit$flag == 1000 && iter.max > 1)
        warning("Ran out of iterations and did not converge")
if(coxfit$flag < 0)
```

```
        stop(paste("X matrix deemed to be singular; variable",  -
                coxfit$flag))
if(coxfit$flag > 0 & coxfit$flag <= nvar) {
        stop(paste("Variable ", coxfit$flag, "is becoming infinite:",
                coxfit$coef[coxfit$flag]))
}
rownames.x <- dimnames(x)[[1]]
colnames.x <- dimnames(x)[[2]]
if(length(colnames.x) > 0)
        names(coxfit$coef) <- colnames.x
if(table.n)
        retlist <- list(n = n.table, coef = coxfit$coef, var = matrix(
                coxfit$imat, ncol = nvar), loglik = coxfit$loglik,
                score = coxfit$sctest, iter = coxfit$iter)
else retlist <- list(coef = coxfit$coef, var = matrix(coxfit$imat, ncol
                 = nvar), loglik = coxfit$loglik, score = coxfit$sctest,
                iter = coxfit$iter)     #
attr(retlist, "class") <- "coxreg"
if(resid.int == 1)
        return(retlist) # none to do!
score <- as.double(exp(xx %*% coxfit$coef) * wt[sorted])
coxhaz <- .C("coxhaz",
        as.integer(nused),
        score,
        coxfit$mark,
        newstrat,
```

```
            hazard = double(nused),
            cumhaz = double(nused))
    if(resid.int == 2) {
#martingale residuals
            resid <- rep(NA, n)
            resid[sorted] <- sstat - score * coxhaz$cumhaz
            if(length(rownames.x) > 0)
                    names(resid) <- rownames.x
            retlist$resid <- resid
            return(retlist)
    }
    if(resid.int == 3) {
#deviance residuals
            resid <- rep(NA, n)
            temp <- sstat - score * coxhaz$cumhaz
            temp2 <- sstat * log(ifelse(temp == 0, 1, score * coxhaz$cumhaz
                    ))
            temp2 <- sqrt(-2 * (temp + temp2))
            resid[sorted] <- ifelse(temp < 0,  - temp2, temp2)
            if(length(rownames.x) > 0)
                    names(resid) <- rownames.x
            retlist$resid <- resid
            return(retlist)
    }
    if(resid.int == 4) {
# score residuals
```

```
            temp <- .C("coxres1",
                    as.integer(nused),
                    as.integer(nvar),
                    stime,
                    sstat,
                    xx,
                    newstrat,
                    score,
                    coxhaz$hazard,
                    coxhaz$cumhaz,
                    resid = double(nused * nvar),
                    wmean = double(nused * nvar))
            temp <- matrix(temp$resid, ncol = nvar)
            resid <- matrix(NA, ncol = nvar, nrow = n)
            dimnames(resid) <- dimnames(x)
            resid[sorted, ] <- temp
            retlist$resid <- drop(resid)
            return(retlist)
        }
        if(resid.int == 5) {
#Schoenfeld residuals
            temp <- .C("coxres2",
                    n = as.integer(nused),
                    as.integer(nvar),
                    indx = stime,
                    sstat,
```

```
                        xx,
                        newstrat,
                        score,
                        resid = double(nused * nvar),
                        double(nused * nvar))
                indx <- temp$indx[1:(temp$n)]   #the unique death times
                if(missing(strata)) {
                        resid <- matrix(temp$resid, ncol = nvar)[1:(temp$n),  ,
                                drop = F]
                }
                else {
#put the resids in time order, rather than time within strata
                        ord <- order(stime[indx])
                        indx <- indx[ord]
                        ord <- (1:(temp$n))[ord]
                        resid <- matrix(temp$resid, ncol = nvar)[ord,  , drop
                                = F]
                        retlist$strata <- (strata[sorted])[indx]
                }
                if(length(colnames.x) > 0)
                        dimnames(resid) <- list(NULL, colnames.x)
                retlist$resid <- drop(resid)
                retlist$time <- stime[indx]
                return(retlist)
        }
}
```

Appendix B. Data

B.1 Hughes Missile System Company Production Reliability Acceptance Test Sampling Data

As noted in section 4.2, the captive-carry lifelength observations in the PRAT sampling data are based on FAST results. In order to determine equivalent captive-carry lifelengths based on BIT results, the PRAT results are reviewed in accordance with the following criteria: a missile fails PRAT if a type II BIT failure (i.e., two consecutive BIT failures) is detected during a missile's test sequence. This revised criteria is chosen based on its closeness to the operational failure assessment; recall (refer to section 3.3) that a missile is considered to have failed if any two consecutive BIT failures indicate a failed missile. Consequently, a new PRAT sampling data set is created based on this revised criteria. Below is a review of the HMSC PRAT sampling data based on a **BIT** assessment. The recommendations shown for each sublot were implied. *Test hours include vibration only test time.*

Hughes Lot 2 Sublot 3:

RTV-26 (CA00278): Failed incoming 5-sec BIT. Was not included in the captive flight test simulation scoring.

RTV-27 (CA00279): "No wake up" failure was detected and prevented a successful BIT or FAST from being completed. The missile did not respond to BIT test during cycle 6, BIT 3 (22.37 test hours). Missile did not undergo a FAST. Missile was scored as a failure with 21.50 credited test hours.

RTV-28 (CA00280): Missile passed CFTS undergoing 37 cycles (150.10 test hours). The missile was FASTed in the tactical mode because of suspected problems with the telemetry unit. Testing in the tactical mode bypasses the telemetry unit using a telemetry simulator. The missile failed for "HPRF bird VCXO". Missile was scored a pass in the Naval Air Warfare Center test report, but scored as a failure on Mr. Guglielmoni's PRAT Summary spread sheet.

RTV-29 (CA00281): Missile failed CFTS when a type II BIT (Transmitter (XMTR)) failure was detected during cycle 38, BIT 4 (153.10 test hours). Three type I BITs occurred: Cycle 29, BIT 5 (117.64 test hours) - PRF/PDI CONTROL; Cycle 31, BIT 4 (124.7 test hours) - Seeker Position Mode; Cycle 31, BIT 5 (126.20 test hours) VCXO INIT. The FAST was aborted to prevent further damage to the missile. Missile was scored as a failure with 152.64 credited test hours.

RTV-30 (CA00282): Missile passed CFTS undergoing 43 cycles (174.44 test hours). The missile has 8 type I BIT failures:

- 5 seeker position modes failures. - (Cycle 12, BIT 1 (44.71 test hours); Cycle 12, BIT 2 (44.98 test hours); Cycle 14, BIT 1 (52.82 test hours); Cycle 43, BIT 4 (173.38 test hours).
- SKR POS MODE - Cycle 15, BIT 2 (57.15 test hours).
- SKR RATE MODE - Cycle 22, BIT 2 (89.60 test hours).
- SEEKER BORESIGHT- Cycle 39, BIT 2 (158.57 test hours).

During the initial FAST, a guidance section temperature sensor interlock prevented testing due to a failure of the GS1 temperature sensor. The GS1 temperature sensor was disabled to allow FAST to continue. The missile was tested in the normal PRAT mode and in the tactical mode. It failed the seeker servo position linearity test in both configurations. The missile was scored as a failure.

RTV-31 (CA00283): Missile passed CFTS undergoing 32 cycles (129.81 test hours). No BIT failures were detected. FAST was performed twice. It first failed several range measurements which were due to a faulty low frequency counter in the FAST. It also failed the "GS 15 volt power" which is an intermittent test equipment cabling problem. After replacing the low frequency counter the FAST retest passed. The missile was scored as a pass.

RTV-32 (CA00284): Missile failed CFTS when a type II BIT (Target Detection Device (TDD)) failure was detected during cycle 2, BIT 4 (7.06 test hours). The missile passed FAST in

both the normal PRAT and the tactical mode. TDD failure was also detected at the contractor's facility. The missile was scored a pass with 6.60 credited test hours.

RTV-33 (CA00285): Failed incoming 5-sec BIT. Was not included in the captive flight test simulation scoring.

RTV-34 (CA00286): Missile failed CFTS when a type II BIT (High Pulse Repetition Frequency (HPRF) mode) failure was detected during cycle 11, BIT 4 (43.58 test hours). During FAST, the missile failed the HPRF image, HPRF sensitivity, and several filter processor tests. The missile was scored a failure with 43.11 credited test hours.

RTV-35 (CA00287): Missile failed CFTS when a type I BIT (PRF/PDI control) failure was detected during cycle 6, BIT 3 (22.37 test hours). The missile has multiple Filter Processor/Range Correlator (FP/RC) synchronization failures during the 3-sec CFTS BITS. The All-Up-Round (AUR) 5-sec BIT failed seeker rate, seeker position, and seeker boresight. When FAST was performed, the test aborted due to antenna elevation and azimuth position. Two independent failure modes were discovered on this missile, the filter processor and the seeker servo. The PRAT committee counted the CFTS failure time to the first failure so it was counted as a single missile failure. The missile was scored a failure with 21.50 credited test hours.

Summary: 10 missiles were tested

- 2 missiles (RTV-26 and RTV-33) failed incoming 5-sec BIT and where not counted in the scoring.
- 8 data points
  - 6 missiles (RTV-27,28,29,30,34,35) were scored as failures
  - 2 missiles (RTV-31, 32) were scored as passes (probably should have credited RTV-32 with time up to cycle 2 BIT 4 since was not a failure)

Recommendation:

- RTV-32 should be scored as a failure rather than a pass based on CFTS failure indication.
- RTV-28 and RTV-30 should be scored as passes rather than failures based on no CFTS failure. indications.

Hughes Lot 3 Sublot 1:

RTV-01 (CA00837): Missile passed CFTS undergoing 47 cycles (190.66 test hours). The missile had one type I BIT Failure (HPRF mode) at cycle 9, BIT 5 (36.50 test hours). The missile was FASTed in the tactical mode because of suspected problems with the telemetry unit. The missile passed FAST and was scored as a pass.

RTV-02 (CA00838): Missile was credited with passing CFTS undergoing 32 cycles (129.81 test hours). However, the missile had five Type I and two Type II BIT failures for seeker position mode. The first BIT failure was a type II at cycle 15, BIT 4 (59.793 test hours). The missile passed FAST. Thus, the BIT appeared to be an intermittent failure mode. The missile was scored as a pass.

RTV-03 (CA00839): Missile passed CFTS undergoing 37 cycles (150.10 test hours). The missile has three type I BIT failures: Cycle 30, BIT 3 (119.73 test hours) - first ACG INT then HPRF mode; Cycle 36, BIT 2 (142.34 test hours) - ACG INT. Missile passed FAST and was scored as a pass.

RTV-04 (CA00840): Missile failed CFTS when a type II (Bit:AGC INT, TGT Image; ReBit: AGC INT, VCXO INIT, HPRF MODE, TGT IMAGE) failure was detected during cycle 10, BIT 2 (36.87 test hours). Missile failed FAST for multiple IF receiver tests including MPRF and HPRF signal to noise ration. Missile was scored as a failure with 36.73 credited hours.

RTV-05 (CA00841): Missile failed CFTS when a type II (Bit: AGC INT, RANGE INT, RF ATTEN #1, AGC COMM, DAGC COMM, RANGE COMM, HPRF MODE, PNCODE, ENSEMBLE AVG SUM GUARD MODE; ReBit: AGC INT, VCXO INT, RANGE INT, AGC INT, VCXO INT, AGC COMM, PN CODE, ENSEMBLE COUNTER, DC OFFSET) failure was de-

tected during cycle 12, BIT 4 (47.63 test hours). A type I failure (PRF/PDI CONTROL) was detected during cycle 8, BIT 4 (31.40 test hours). On the first FAST attempt, the Automatic Gain Control (ACG) initialization failure mode seen during CFTS was confirmed. However, the FAST test aborted before completion because the transmitter shut down due to a test equipment problem. On the second attempt, the FAST aborted for the same reason. The third FAST attempt was successful. The missile was returned to the HAC for failure analysis. At the time of test report writing, HAC had not isolated the cause of the failure mode. The missile was scored as a failure with 47.17 credited test hours.

RTV-06 (CA00842): The missile passed CFTS undergoing 37 cycles (150.10 test hours). The missile had three type I BIT failures during CFTS: Cycle 15, BIT 3 (58.88 test hours) - HPRF mode, Cycle 29, BIT 4 (116.59 test hours) - AGC INIT, Cycle 36, BIT 2 (142.34) - AGC COMM; RF ATTEN 1, RF ATTEN 2. The missile passed FAST and was scored a pass.

RTV-07 (CA00843): The missile passed CFTS undergoing 41 cycles (166.32 test hours). No BIT failures occurred. The missile passed FAST and was scored a pass.

RTV-08 (CA00844): The missile passed CFTS undergoing 41 cycles (166.32 test hours). No BIT failures occurred. The missile failed two FAST attempts due to seeker position linearity. This failure mode is not BIT detectable. The failure was classified as non-relevant since it is probably due to a high friction problem that would have no mission impact. Based on the FAST data, additional FAST attempts would most likely pass. The missile was scored a pass.

RTV-14 (CA00850): The missile passed CFTS undergoing 32 cycles (129.81 test hours). No BIT failures occurred. The missile passed FAST and was scored a pass.

RTV-15 (CA00851): The missile failed CFTS when a type II (Bit: DAGC INIT, AGC INIT, PHASE INIT, RANGE INIT, VCXO INIT, DAGC COMM, AGC COMM, RANGE COMM, VCXO LIMITS, HPRF MODE, FRU P.L.L., PN CODE, TGT IMAGE, RDI FAIL, INT RCVR NOISE, RANGE RATE) failure was detected during cycle 5, BIT 3 (18.31 test hours). The missile had one

type I failure (VCXO INIT) at cycle 4, BIT 3 (14.25 test hours). All the SRTS failures occurred during the third BIT of cycles at temperature offsets of 0,-6, and 5 degrees F. The missile failed the first two FAST attempts for Guidance Section (G/S) -15 Volt Standard Deviation. This failure mode was isolated to a faulty telemetry unit on a subsequent FAST in tactical mode. A fourth FAST was attempted and the LCT aborted. A fifth FAST attempt was successful. The missile was scored a failure with 17.45 credited test hours.

Summary:

- 10 missiles were tested
- 10 missiles were used as data points
  - 7 missiles passed (RTV-1,2,3,6,7,8,14)
  - 3 missiles failed (RTV-4,5,15)

Recommendation:

RTV-2 should be recorded as a failure at the first type II BIT failure.

Hughes Lot 3 Sublot 2: (Information is not from Test Report)

RTV-16 (CA00852): The missile failed CFTS when the second type II BIT (Bit: Phase INIT, PN Codes; ReBit: PN Codes) failure was detected at cycle 17, BIT 1 (64.99 test hours). The missile had two type I BIT failures (PHASE INIT, PN CODES - Cycle 5, BIT 5 (20.28 test hours); PN CODES - Cycle 6, BIT 5 (24.34 test hours)) and one type II failure (PN Codes - Cycle 7, BIT 1 (24.42 test hours). The missile failed two FASTs with PN CODES and HI PN PULSE WIDTH. The missile was scored as a failure with 64.95 credited test hours.

RTV- 17 (CA00852): The missile passed CFTS undergoing 51 cycles (206.89 test hours). The missile had one type I failure (Seeker Position Mode) at cycle 17, BIT 3 (66.99 test hours). The missile was FASTED in the tactical mode because of suspected telemetry problems and passed. The missile was scored a pass.

RTV- 18 (CA00853): The missile passed CFTS undergoing 45 cycles (182.55 test hours). The missile had no BIT failures. FAST indicated an RDL external during BIT. The missile was scored a pass.

RTV- 19 (CA00854): The missile passed CFTS undergoing 34 cycles (137.93 test hours). The missile had no BIT failures and passed FAST. The missile was scored a pass.

RTV- 20 (CA00856): The missile failed CFTS when a type II BIT (Bit: TDD, Rebit: TDD) failure occurred during cycle 5, BIT 3 (18.31 test hours). The missile failed the initial FAST with a TDD Doppler Frequency caused by a bad oscillator/mixer in station TDD repeater. The missile passed the second FAST. The missile was scored a failure with 17.45 credited test hours.

RTV- 21 (CA00857): The missile failed CFTS when a type II BIT (Bit: HPRF mode, ReBit: HPRF mode) failure occurred at cycle 15, Bit 4 (59.79 test hours) . Three type I BIT failures (also HPRF mode) occurred before this type II failure (Cycle 8, BIT 4 - 34.45 test hours; Cycle 12, BIT 4 - 51.68 test hours; Cycle 13, BIT 4 - 55.74 test hours). The missile failed two FASTs with TDD RF power. TDD RF power is not BIT detectable. TDD RF Power and HPRF Target to Image Ratio are not related. Two separate failure modes occurred. The missile was scored a failure with 59.34 credited test hours.

RTV- 22 (CA00858): The missile passed CFTS undergoing 55 cycles (223.12 test hours). The missile has one type I BIT failure (Spare - RDI continuity) at cycle 47, BIT 4 (189.61 test hours). The missile passed two FASTs and was scored a pass.

RTV- 25 (CA00861): The missile passed CFTS undergoing 55 cycles (223.12 test hours). No BIT failure were detected.). The missile passed two FASTs and was scored a pass.

RTV- 26 (CA00862): The missile passed CFTS undergoing 44 cycles (178.49 test hours). The missile had one type I BIT failure (PRF/PDI Control) at Cycle 8, BIT 1 (28.48 test hours). The missile passed a FAST and was scored a pass.

RTV- 27 (CA00863): The missile passed CFTS undergoing 44 cycles (178.49 test hours). No BIT failures were detected. The missile passed a FAST and was scored a pass.

Summary:

- 10 missiles were tested
- 10 missiles were used as data points
  - 7 missiles passed (RTV- 17,18,19,22,25,26,27)
  - 3 missiles failed (RTV- 16,20,21)

Recommendation:

RTV-16 failure should be recorded at the first type II BIT failure (Cycle 7, BIT 1 - 24.42 test hours) instead of the second.

Hughes Lot 4 Sublot 1: was not able to get a BIT report.

RTV-01 (CA01275): The missile passed CFTS undergoing 42 cycles (170.38 test hours). No BIT failures were detected. The missile passed a FAST and was scored a pass.

RTV-02 (CA01276): The missile passed CFTS undergoing 83 cycles (336.70 test hours). No BIT failures were detected. The missile passed a FAST after 56 cycles and another FAST after 27 cycles (incentive). The missile was scored a pass.

RTV-04 (CA01278): The missile passed CFTS undergoing 83 cycles (336.70 test hours). No BIT failures were detected. The missile passed a FAST after 56 cycles and another FAST after 27 cycles (incentive). The missile was scored a pass.

RTV-05 (CA01279): The missile passed CFTS undergoing 42 cycles (170.38 test hours). No BIT failures were detected. The missile passed a FAST and was scored a pass.

RTV-06 (CA01280): The missile passed CFTS undergoing 42 cycles (170.38 test hours). Two type I BIT (SEEKER RATE MODE and SEEKER POSITION) failures were detected at Cycle 1, BIT 5 (4.5 test hours). The missile passed a FAST and was scored a pass.

RTV-07 (CA01281): The missile passed CFTS undergoing 42 cycles (170.38 test hours). No BIT failures were detected. The missile passed a FAST and was scored a pass.

RTV-08 (CA01282): The missile passed CFTS with 34 cycles (137.93 test hours). However, the missile had a type II BIT (Bit: Actuator, ReBit: Actuator) failure at cycle 6, BIT 5 (24.34 test hours). The missile also had a type I BIT (TDD) at cycle 8, BIT 3 (30.48 test hours). The missile failed the first three FASTs for Actuator #1 Zero Offset Locked, but passed the fourth FAST. The test results improved with each successive test (possible actuator mechanical problem that improves with exercising. The missile was scored a pass.

RTV-09 (CA01283): The missile failed CFTS when a type II BIT (Bit: TDD, ReBit: TDD) failure at cycle 12, BIT 3 (46.71 test hours). The missile passed FAST, but was scored a failure with 45.84 credited test hours based on eight type II BIT (TDD) failures. All failures occurred in the coldest BIT, BIT 3 (Failures occured at GS Temperatures of -10 to -17 degrees C).

RTV-10 (CA01284): The missile passed CFTS undergoing 34 cycles (137.93 test hours). The missile had one type I BIT (Seeker Rate Mode) failure at cycle 22, BIT 3 (87.27 test hours). The missile failed three FASTs for ADP Test Flag 1B and the flag caused FAST to be aborted. This failure would not affect launch. The missile was shipped to HMSC where it passed FAST. The missile was scored a pass.

RTV-11 (CA01285): The missile failed CFTS when a type II BIT (Bit: DAGC INIT, ReBit: DAGC INIT) failure at cycle 18, BIT 4 (71.96 test hours). The missile passed FAST, but was scored a failure with 71.51 credited test hours based on fourteen type II BIT (DAGC INIT) failures. The failures appear temperature induced (probably faulty frequency reference unit).

RTV-12 (CA01286): The missile passed CFTS undergoing 34 cycles (137.93 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

RTV-13 (CA01287): The missile passed CFTS undergoing 34 cycles (137.93 test hours). One type I BIT (SPARE - RDI Continuity) failure was detected at cycle 13, BIT 2 (49.04 test hours). The missile passed FAST and was scored a pass.

Summary:

- 12 missiles were tested
- 12 missiles were used as data points
  - 10 missiles passed (RTV- 1,2,4,5,6,7,8,10,12,13)
  - 2 missiles failed (RTV- 9,11)

Recommendation:

RTV-8 be recorded as a failure due to type II BIT failure indication (cycle 8, BIT 3 - 30.48 test hours).

Hughes Lot 4, Sublot 2:

RTV-14 (CA01288): The missile passed CFTS undergoing 72 cycles (292.08). However, the missile had one type II BIT (Bit and ReBit: RDI Continuity (1553) No INRT 8 Multiple Fails TM) failure at cycle 48, BIT 3 (192.75 test hours). The missile was FASTed after 53 cycles and passed. The missile was FASTed after another 19 cycles and passed. The missile was scored a pass.

RTV-15 (CA01289): The missile passed CFTS undergoing 42 cycles (170.38). No BIT failures were detected. The missile passed FAST and was scored a pass.

RTV-16 (CA01290): The missile passed CFTS undergoing 42 cycles (170.38). No BIT failures were detected. The missile passed FAST and was scored a pass.

RTV-17 (CA01291): The missile failed CFTS when a type II BIT (Bit and ReBit: TDD) failure was detected at Cycle 1, BIT 3 (2.08 test hours). The missile passed FAST in tactical mode and passed. The missile was scored a failure with 1.22 credited test hours based on seven type II BIT (TDD) failures. These failures occured during BIT 3 in cycles with the colder temperature offset.

RTV-18 (CA01292): The missile passed the incoming 5-sec BIT, but failed the 3-sec pre-environmental BIT for Frequency Reference Unit (FRU) Lock Loop. However, the missile was not scored.

RTV-19 (CA01293): The missile passed CFTS undergoing 41 cycles (166.32 test hours). No BIT failures were detected. The missile passed FAST in tactical mode (generally due to telemetry problems). The missile was scored a pass.

RTV-20 (CA01294): The missile passed CFTS undergoing 70 cycles (283.97 test hours). No BIT failures were detected. The missile was FASTed after 51 cycles and passed. The missile was FASTed after another 19 cycles and passed. The missile was scored a pass.

RTV-21 (CA01295): The missile passed CFTS undergoing 70 cycles (283.97 test hours). No BIT fails were detected. The missile was FASTed after 51 cycles and passed. The missile was FASTed after another 19 cycles and failed for high -110 volt control section current. The measured value was 25 amps while the expected upper limit for FAST is only 0.7 amps. The test set applies power after a simulated squib fire in the launch control test. It waits for power to settle (0.75 secs) and then measures the current. The control section power was removed immediately by the test equipment. While attempting to retest, the test aborted in BIT when attempting the actuator tests. The failure is not BIT detectable. The missile was scored a pass, but with only 51 cycles (206.89 test hours). Note: 60.5 cycles (using 1/2 of 19 cycles) was used for MBTF scoring - 245.43 credited test hours.

RTV-22 (CA01296): The missile passed CFTS undergoing 51 cycles (206.89 test hours). No BIT failures were detected. The missile failed FAST for the seeker elevation 10 second drift test. The measured value was 1.56 degrees. The upper limit is 0.42 degrees. During a second FAST attempt, the missile again failed the drift test with a measured value of 0.73 degrees. The missile failed the actuator YAW DAC test during both FAST attempts. Test personnel believed the actuator YAW DAC failures are not relevant and due to telemetry data shifts. Engineering groups determined the FAST failure modes would not affect the mission performance of the missile. Since the failures were considered non-relevant, the missile was scored a pass.

RTV-23 (CA01297): The missile passed CFTS undergoing 41 cycles (166.32 test hours). No BIT failures were detected. The missile failed test and retest for Actuator Roll DAC. Both failures were over 70 counts. Would expect multiple failures if DAC error was actually this high. Probably telemetry unit problem, but can not verify by testing in tactical mode because C/S DAC tests are not performed in the tactical mode. Missile was scored as a pass.

RTV-24 (CA01298): The missile passed CFTS undergoing 49 cycles (198.78 test hours). No BIT failures were detected. The missile passed FAST and was scored as a pass.

RTV-25 (CA01299): The missile passed CFTS undergoing 31 cycles (125.76 test hours). No BIT failures were detected. The missile passed FAST and was scored as a pass.

Summary: 12 missiles were tested

- 1 missile failed incoming BIT (RTV-18)
- 11 missiles were used as data points
  - 9 missiles passed (RTV- 14,15,16,19,20,22,23,24,25)
  - 2 missiles failed (RTV- 17,21)

Recommendation:

- RTV-14 should be reported as a failure based type II BIT failure at cycle 48, BIT 3 (192.75 test hours).
- RTV-18 should be reported as a failure based on 3-sec BIT identifying failure (0 test hours).
- RTV-21 should be reported as a pass based on no BIT failures (70 cycles - 283.97 test hours).

Hughes Lot 5, Sublot 1: FAST is now conducted at HMSC.

RTV-01 (CA01815): While installing the missile into the test chamber, test personnel discovered loose hardware inside the missile's telemetry section (S/N 1002019). Since the telemetry section is GFE, the missile was scored as a not test against the PRAT requirements.

RTV-02 (CA01816): The missile passed CFTS undergoing 42.07 cycles (170.66 test hours). No BIT failures were detected. The missile passed fast and was scored a pass.

RTV-03 (CA01817): The missile failed CFTS when a type II BIT (Bit & ReBit: Max Current Draw - Deselected Missile; drew excess 400 Hz current during BIT) at cycle 4, BIT 3 (14.25 test hours). Failure was detected when STRS was unable to load umbilical simulator; a manual bit was performed. This type of failure is internal to the missile and not caused by external power source. The missile was returned to HMSC where its guidance section passed a FAST. The missile failure was later isolated to the filter rectifier (P/N 7005325, S/N 5087). Failure analysis revealed an electrical short from -135 volts DC to the chassis. There was a pinched wire between the housing and the hybrid case. The root cause was workmanship. The missile was scored a failure with 13.39 credited test hours.

RTV-04 (CA01818): The missile passed CFTS undergoing 46 cycles and 16.8 minutes (186.89 test hours). The missile pass FAST and was scored as a pass.

RTV-05 (CA01819): The missile failed CFTS when a type II BIT (Bit: DAGC INIT, AGC INIT, PHASE INIT, RNG INIT, VCXO INIT, DAGC CMD, AGC CMD, RF ATTEN #1 & 2, RNG CMD, HPRF MODE, PN CODE, TGT IMAGE, MIN NOISE, RDI FAIL, INT DATA LINK,

RNG RATE; ReBit: Most the same failure modes for retest) failure at cycle 22, BIT 4 (88.19 test hours). The missile failed DAGC initialization. This failure was confirmed by FAST. The failure analysis had not been completed but the contractor suspects a failure with a local oscillator. The missile was scored a failure with 87.73 credited test hours.

RTV-06 (CA01820): The missile passed CFTS undergoing 65 cycles (263.68 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

RTV-07 (CA01821): The missile passed CFTS undergoing 70 cycles (283.97 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

RTV-08 (CA01822): The missile passed CFTS undergoing 45 cycles (182.55 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

RTV-09 (CA01823): The missile passed CFTS undergoing 45 cycles (182.55 test hours). A type I BIT (Seeker Position Mode) failure was detected at cycle 24, BIT 4 (96.30). The missile passed FAST and was scored a pass.

RTV-10 (CA01824): The missile passed CFTS undergoing 45 cycles (182.55 test hours). However, the missile had two type I BIT (Cycle 9, BIT 3 (34.54 test hours) - RNG COMM, VCXO LIMITS, MPRF MODE, HPRF MODE, PRF/PDI CONT, PN CODE, DC OFFSETS, RDI FAIL, INT DATA LINK, ENS CNTR, BIRD O & -10 DB; Cycle 23, BIT 4 (92.25 test hours) - RANGE COMMAND) failures and two type II BIT (Cycle 16, BIT 1 (60.93 test hours) - Bit: AGC, RNG & VCXO INIT, AGC CMD, RF ATTEN #1, RNG COMM, HPRF MODE, FRU PLL CODE, RDI FAIL, INT RCVR NOISE, DC OFFSET (-10), RNG RATE & ReBit: Add PHASE INIT & subtract RF ATTEN #1; Cycle 16, BIT 2 (61.21 test hours) - Bit: AGC CMD, RF ATTEN #1, HPRF MODE, PN CODE, DC OFFSETS, DC OFFSET (-10) & ReBit: AGC, PHASE, RNG & VCXO INIT, RNG CMD, FRU PLL, RDI FAIL, INT RCVR NOISE, RANGE RATE) failures. Missile passed FAST and was scored a pass based on 89 additional hours without another failure.

RTV-11 (CA01825): The missile failed CFTS when a type II BIT (Bit & ReBit: Internal Data Link) failure was detected at cycle 3, BIT 4 (11.11 test hours). Five additional type II BITs ware recorded. Missile failed BIT due to a low reading for the internal rear data link noise filter pass 1. The missile passed FAST and ambient BITs at HMSC. The failure was later duplicated at cold temperature. The missile was scored a failure with 10.66 credited test hours due to Type II criteria.

RTV-12 (CA01826): The missile passed CFTS undergoing 45 cycles (182.55 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

Summary: 12 missiles were tested

- 1 missile was not scored due to a GFE problem
- 11 missiles were used as data points
  - 8 missiles passed (RTV- 2,4,6,7,8,9,10,12)
  - 3 missiles failed (RTV- 3,5,11)

Recommendation:

RTV-10 should be recorded a failure based on 2 type II BIT failures, first seen at Cycle 16, BIT 1 (60.93 test hours).

Hughes Lot 5, Sublot 2:

RTV-13 (CA01827): The missile failed incoming 5-sec BIT (Actuator) and was not counted in the CFTS scoring.

RTV-14 (CA01828): The missile passed CFTS undergoing 57 cycles (231.23 test hours). No BIT failures were detected. The missile was scored as a pass.

RTV-15 (CA01829): The missile passed CFTS undergoing 57 cycles (231.23 test hours). No BIT failures were detected. The missile failed FAST at HMSC for Seeker Servo Settling Time

(failed rate step 1 settling time in AZ & EL). The failure was not seen when the missile skin was removed. Cause was traced to workmanship; out of tolerance bend for upper antenna baffle. Baffle was rubbing on radome. The missile was scored a failure and was credited with half time, 28.5 cycles (115.62 test hours).

RTV-16 (CA01830): The missile passed CFTS undergoing 23 cycles (93.30 test hours). No BIT failures were detected. The missile was sent to China Lake for Fixture/Shaker validation. The missile was scored a pass.

RTV-17 (CA01831): The missile passed CFTS undergoing 57 cycles (231.23 test hours). No BIT failures were detected. The missile was scored as a pass.

RTV-18 (CA01832): The missile passed CFTS undergoing 23 cycles (93.30 test hours). No BIT failures were detected. The missile was sent to China Lake for Fixture/Shaker validation. The missile was scored a pass.

RTV-19 (CA01833): The missile passed CFTS undergoing 57 cycles (231.23 test hours). No BIT failures were detected. The missile failed FAST for MPRF FUF Mode, but passed 2 subsequent FASTs. The missile was scored as a pass.

RTV-20 (CA01834): The missile passed CFTS undergoing 55 cycles (223.12 test hours). A type I BIT (RDI Continuity) failure was detected at cycle 50, BIT 5 (203.28 test hours). The missile was scored as a pass.

RTV-21 (CA01835): The missile passed CFTS undergoing 43 cycles (174.44 test hours). No BIT failures were detected. The missile was scored as a pass.

RTV-22 (CA01836): The missile passed CFTS undergoing 43 cycles (174.44 test hours). No BIT failures were detected. The missile was scored as a pass.

RTV-23 (CA01837): The missile passed CFTS undergoing 55 cycles (223.12 test hours). No BIT failure were detected. The missile was scored as a pass.

RTV-24 (CA01838): The missile failed incoming 5-sec BIT (Actuator) and was not counted in the CFTS scoring.

Summary: 12 missiles were tested

- 2 missiles (RTV-13,24) were not scored as a result of incoming BIT failures.
- 10 missiles were used as data points
  - 9 missiles passed (RTV- 14,16,17,18,19,20,21,22,23)
  - 1 missiles failed (RTV- 15)

Recommendation: RTV-15 should be recorded as a pass since a BIT failure was not detected and given full PRAT time, 57 cycles (231.23 test hours).

Hughes Lot 6, Sublot 1:

RTV-01 (CA02074): The missile passed CFTS undergoing 47 cycles (190.66 test hours). No BIT failures were detected. The missile passed FAST at Tucson and was scored a pass.

RTV-02 (CA02075): The missile passed CFTS undergoing 47 cycles (190.66 test hours). No BIT failures were detected. The missile passed FAST at Tucson and was scored a pass.

RTV-03 (CA02076): The missile passed CFTS undergoing 47.64 cycles (193.28 test hours). No BIT failures were detected. On incoming inspection, the missile had a suspected hardware telemetry problem. The missile was returned to Tucson for telemetry replacement. Upon disassembly, a bent pin was found in the guidance section connector. The pin was straightened and returned to Pt. Mugu for CFTS. The missile passed FAST at Tucson and was scored a pass.

RTV-04 (CA02077): The missile passed CFTS undergoing 39 cycles (158.21 test hours). No BIT failures were detected. The missile passed FAST at Tucson and was scored a pass.

RTV-05 (CA02078): The missile passed CFTS undergoing 50 cycles (202.83 test hours). No BIT failures were detected. The missile passed FAST at Tucson and was scored a pass.

RTV-06 (CA02079): The missile passed CFTS undergoing 50 cycles (202.83 test hours). No BIT failures were detected. The missile passed FAST at Tucson and was scored a pass.

RTV-07 (CA02080): The missile passed CFTS undergoing 39 cycles (158.21 test hours). No BIT failures were detected. The missile passed FAST at Tucson and was scored a pass.

RTV-08 (CA02081): The missile failed CFTS when a type II BIT (Bit: Seeker Position Mode, Seeker Rate Mode; Rebit: Seeker Position Mode, Seeker Rate Mode, Exc Body Motion) failure was detected at cycle 22, BIT 2 (85.50 test hours). The missile intermittently failed during Bit 2 through the remaining 17 cycles. The missile had a total of five type II and three type I failures. The missile passed FAST at Tucson. The missile was scored a failure with 85.41 credited test hours.

RTV-10 (CA02083): The missile passed CFTS undergoing 55 cycles (223.12 test hours). The missile had a type II BIT failure (Bit & ReBit: Actuator) during cycle 23, BIT 2 (89.6 test hours). The missile passed FAST at Tucson and was scored a pass.

RTV-11 (CA02084): The cycle passed CFTS undergoing 42 cycles (170.38 test hours). The missile had three type I BIT failures (Cycle 31, BIT 3 (123.78 test hours) - PN Code; Cycle 36, BIT 1 (142.07 test hours) - FP/RC SYNC; Cycle 37, BIT 4 (149.04 test hours) - AGC INIT). The missile passed FAST at Tucson and was scored a pass.

RTV-12 (CA02085): The missile passed CFTS undergoing 34 cycles (137.93 test hours). No BIT failures were detected. The missile passed FAST at Tucson and was scored a pass.

RTV-13 (CA02086): The missile failed CFTS when a type II BIT failure (Bit & ReBit: Actuator) was detected during cycle 7, BIT 1 (22.98). Note Cycle 6 was only 156.8 minutes in length. The missile failed FAST for Actuator. The missile was scored a failure with 22.94 credited test hours.

Summary: 12 missiles were tested and 12 missiles were used as data points

- 10 missiles passed (RTV- 1,2,3,4,5,6,7,10,11,12)
- 2 missiles failed (RTV- 12,13)

Recommendation:

RTV-10 should be recorded as a failure based on type II BIT failure during cycle 23, BIT 2 (89.6 test hours).

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
RTV-26	H2-3	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
RTV-27	H2-3	1	C6,B3	21.50	27.59	1	C6,B3	21.50	27.59
RTV-28	H2-3	1	C37	150.10	193.26	0	C37	150.10	193.26
RTV-29	H2-3	1	C38,B4	152.64	196.18	1	C38,B4	152.64	196.18
RTV-30	H2-3	1	C43	174.44	224.60	0	C43	174.44	224.60
RTV-31	H2-3	0	C32	129.81	167.15	0	C32	129.81	167.15
RTV-32	H2-3	0	C2,B4	6.60	8.14	1	C2,B4	6.60	8.14
RTV-33	H2-3	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
RTV-34	H2-3	1	C11,B4	43.11	55.15	1	C11,B4	43.11	55.15
RTV-35	H2-3	1	C6, B3	21.50	27.59	1	C6,B3	21.50	27.59
		6		699.70	899.66	5		699.70	899.66
			$\hat{\lambda}$	116.62	149.94		$\hat{\lambda}$	139.94	179.93

Table 9. HSMC Lot 2 Sublot 3 Sampling Data

Below are the sampling data sets used for the HSMC PRAT analysis[1]. The abbreviations are as follows:

- *FAST* - results based on FAST criteria (0 - right censored, 1 - uncensored).
- *BIT* - results based on BIT criteria (0 - right censored, 1 - uncensored).
- *C,B* - Cycle, BIT (Cycle and bit the missile failed a bit on. For example, C6,B3 indicates that the missile was considered to have failed on the third bit of the sixth cycle. If no bit is indicated, then the missile passed PRAT. For example, C37 indicates that the missile passed PRAT undergoing 37 cycles.).
- *Vib Hrs* - Vibration Only test time.
- *Total Hrs* - Total test time.
- $\hat{\lambda}$ - Maximum likehood estimate of the parameter of the Exponential distribution.

[1]Failure times are recorded at the midpoint between BIT checks

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
RTV-01	H3-1	0	C47	190.66	245.50	0	C47	190.66	245.50
RTV-02	H3-1	0	C32	129.81	167.15	1	C15,B4	59.34	76.04
RTV-03	H3-1	0	C37	150.10	193.26	0	C37	150.10	193.26
RTV-04	H3-1	1	C10,B2	36.73	47.36	1	C10,B2	36.73	47.36
RTV-05	H3-1	1	C12,B4	47.17	60.37	1	C12,B4	47.17	60.37
RTV-06	H3-1	0	C37	150.10	193.26	0	C37	150.10	193.26
RTV-07	H3-1	0	C41	166.32	214.16	0	C41	166.32	214.16
RTV-08	H3-1	0	C41	166.32	214.16	0	C41	166.32	214.16
RTV-14	H3-1	0	C32	129.81	167.15	0	C32	129.81	167.15
RTV-15	H3-1	1	C5,B3	17.45	22.36	1	C5,B3	17.45	22.36
		3		1184.47	1524.72	4		1113.99	1433.62
			$\hat{\lambda}$	394.82	508.24		$\hat{\lambda}$	278.50	358.40

Table 10. HSMC Lot 3 Sublot 1 Sampling Data

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
RTV-16	H3-2	1	C17,B1	64.95	83.62	1	C7,B1	24.38	31.38
RTV-17	H3-2	0	C51	206.89	266.39	0	C51	206.89	266.39
RTV-18	H3-2	0	C45	182.55	235.05	0	C45	182.55	235.05
RTV-19	H3-2	0	C34	137.93	177.59	0	C34	137.93	177.59
RTV-20	H3-2	1	C5,B3	17.45	22.36	1	C5,B3	17.45	22.36
RTV-21	H3-2	1	C15,B4	59.34	76.04	1	C15,B4	59.34	76.04
RTV-22	H3-2	0	C55	223.12	287.28	0	C55	223.12	287.28
RTV-25	H3-2	0	C55	223.12	287.28	0	C55	223.12	287.28
RTV-26	H3-2	0	C44	178.49	229.83	0	C44	178.49	229.83
RTV-27	H3-2	0	C44	178.49	229.83	0	C44	178.49	229.83
		3		1472.32	1895.28	3		1431.75	1843.04
			$\hat{\lambda}$	490.77	631.76		$\hat{\lambda}$	477.25	614.35

Table 11. HSMC Lot 3 Sublot 2 Sampling Data

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
RTV-01	H4-1	0	C42	170.38	219.38	0	C42	170.38	219.38
RTV-02	H4-1	0	C83	336.70	433.54	0	C83	336.70	433.54
RTV-04	H4-1	0	C83	336.70	433.54	0	C83	336.70	433.54
RTV-05	H4-1	0	C42	170.38	219.38	0	C42	170.38	219.38
RTV-06	H4-1	0	C42	170.38	219.38	0	C42	170.38	219.38
RTV-07	H4-1	0	C42	170.38	219.38	0	C42	170.38	219.38
RTV-08	H4-1	0	C34	137.93	177.59	1	C8,B3	29.62	38.03
RTV-09	H4-1	1	C12,B3	45.84	58.93	1	C12,B3	45.84	58.93
RTV-10	H4-1	0	C34	137.93	177.59	0	C34	137.93	177.59
RTV-11	H4-1	1	C18,B4	71.51	91.71	1	C18,B4	71.51	91.71
RTV-12	H4-1	0	C34	137.93	177.59	0	C34	137.93	177.59
RTV-13	H4-1	0	C34	137.93	177.59	0	C34	137.93	177.59
		2		2023.98	2605.61	3		1915.67	2466.05
			$\hat{\lambda}$	1011.99	1302.80		$\hat{\lambda}$	638.56	822.02

Table 12. HSMC Lot 4 Sublot 1 Sampling Data

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
RTV-14	H4-2	0	C72	292.08	376.08	1	C48,B3	191.88	246.97
RTV-15	H4-2	0	C42	170.38	219.38	0	C42	170.38	219.38
RTV-16	H4-2	0	C42	170.38	219.38	0	C42	170.38	219.38
RTV-17	H4-2	1	C1,B3	1.22	1.47	1	C1,B3	1.22	1.47
RTV-18	H4-2	N/A	N/A	N/A	N/A	1	C0	0.00	0.00
RTV-19	H4-2	0	C41	166.32	214.16	0	C41	166.32	214.16
RTV-20	H4-2	0	C70	283.97	365.63	0	C70	283.97	365.63
RTV-21	H4-2	1	C60.5	245.43	316.01	0	C70	283.97	365.63
RTV-22	H4-2	0	C51	206.89	266.39	0	C51	206.89	266.39
RTV-23	H4-2	0	C41	166.32	214.16	0	C41	166.32	214.16
RTV-24	H4-2	0	C49	198.78	255.94	0	C49	198.78	255.94
RTV-25	H4-2	0	C31	125.76	161.92	0	C31	125.76	161.92
		2		2027.53	2610.53	3		1965.87	2531.03
			$\hat{\lambda}$	1013.76	1305.26		$\hat{\lambda}$	655.29	843.68

Table 13. HSMC Lot 4 Sublot 2 Sampling Data

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
RTV-01	H5-1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
RTV-02	H5-1	0	C42.07	170.66	219.75	0	C42.07	170.66	219.75
RTV-03	H5-1	1	C4,B3	13.39	17.14	1	C4,B3	13.39	17.14
RTV-04	H5-1	0	C46.07	186.89	240.64	0	C46.07	186.89	240.64
RTV-05	H5-1	1	C22,B4	87.73	112.61	1	C22,B4	87.73	112.61
RTV-06	H5-1	0	C65	263.68	339.52	0	C65	263.68	339.52
RTV-07	H5-1	0	C70	283.97	365.63	0	C70	283.97	365.63
RTV-08	H5-1	0	C45	182.55	235.05	0	C45	182.55	235.05
RTV-09	H5-1	0	C45	182.55	235.05	0	C45	182.55	235.05
RTV-10	H5-1	0	C45	182.55	235.05	1	C16,B1	60.89	78.39
RTV-11	H5-1	1	C3,B4	10.66	13.36	1	C3,B4	10.66	13.36
RTV-12	H5-1	0	C45	182.55	235.05	0	C45	182.55	235.05
		3		1747.18	2248.84	4		1625.52	2092.19
			$\hat{\lambda}$	582.39	749.61		$\hat{\lambda}$	406.38	523.05

Table 14. HSMC Lot 5 Sublot 1 Sampling Data

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
RTV-13	H5-2	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
RTV-14	H5-2	0	C57	231.23	297.73	0	C57	231.23	297.73
RTV-15	H5-2	1	C28.5	115.62	148.87	0	C57	231.23	297.73
RTV-16	H5-2	0	C23	93.30	120.14	0	C23	93.30	120.14
RTV-17	H5-2	0	C57	231.23	297.73	0	C57	231.23	297.73
RTV-18	H5-2	0	C23	93.30	120.14	0	C23	93.30	120.14
RTV-19	H5-2	0	C57	231.23	297.73	0	C57	231.23	297.73
RTV-20	H5-2	0	C55	223.12	287.28	0	C55	223.12	287.28
RTV-21	H5-2	0	C43	174.44	224.60	0	C43	174.44	224.60
RTV-22	H5-2	0	C43	174.44	224.60	0	C43	174.44	224.60
RTV-23	H5-2	0	C55	223.12	287.28	0	C55	223.12	287.28
RTV-24	H5-2	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		1		1791.02	2306.10	0		1906.63	2454.97
			$\hat{\lambda}$	1791.02	2306.10		$\hat{\lambda}$	1906.63	2454.97

Table 15. HSMC Lot 5 Sublot 2 Sampling Data

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
RTV-01	H6-1	0	C47	190.66	245.50	0	C47	190.66	245.50
RTV-02	H6-1	0	C47	190.66	245.50	0	C47	190.66	245.50
RTV-03	H6-1	0	C47.64	193.26	248.84	0	C47.64	193.26	248.84
RTV-04	H6-1	0	C39	158.21	203.71	0	C39	158.21	203.71
RTV-05	H6-1	0	C50	202.83	261.17	0	C50	202.83	261.17
RTV-06	H6-1	0	C50	202.83	261.17	0	C50	202.83	261.17
RTV-07	H6-1	0	C39	158.21	203.71	0	C39	158.21	203.71
RTV-08	H6-1	1	C22,B2	85.41	110.04	1	C22,B2	85.41	110.04
RTV-10	H6-1	0	C55	223.12	287.28	1	C23,B2	89.47	115.26
RTV-11	H6-1	0	C42	170.38	219.38	0	C42	170.38	219.38
RTV-12	H6-1	0	C34	137.93	177.59	0	C34	137.93	177.59
RTV-13	H6-1	1	C5.64,B1	22.92	29.50	1	C5.64,B1	22.92	29.50
		2		1936.43	2493.38	3		1802.78	2321.35
			$\hat{\lambda}$	968.21	1246.69		$\hat{\lambda}$	600.93	773.78

Table 16. HSMC Lot 6 Sublot 1 Sampling Data

Failure Criteria	# of Failures	Vib Hrs	Total Hrs
FAST	22	12882.62	16584.11
	$\hat{\lambda}$	585.57	753.82
BIT	25	12461.91	16041.90
	$\hat{\lambda}$	498.48	641.68

Table 17. HSMC Sampling Data Sets

B.2 Raytheon Company Production Reliability Acceptance Test Sampling Data

As noted in section 4.2, the captive-carry lifelength observations in the PRAT sampling data are based on FAST results. In order to determine equivalent captive-carry lifelengths based on BIT results, the PRAT results are reviewed in accordance with the following criteria: a missile fails PRAT if a type II BIT failure (i.e., two consecutive BIT failures) is detected during a missile's test sequence. This revised criteria is chosen based on its closeness to the operational failure assessment; recall (refer to section 3.3) that a missile is considered to have failed if any two consecutive BIT failures indicate a failed missile. Consequently, a new PRAT sampling data set is created based on this revised criteria. Below is a review of the Raytheon PRAT sampling data based on a **BIT** assessment. The recommendations shown for each sublot were implied. *Test hours include vibration only test time.*

Raytheon Lot 2, Sublot 3:

KE-157 (CA50212): The missile passed CFTS undergoing 27 cycles (109.53 test hours). The missile had 3 type I BIT failures (PROM CHKSUM) during cycle 1, BIT 4 (3.00 test hours); cycle 6, BIT 4 (23.28 test hours); and cycle 23, BIT 5 (93.30 test hours). The missile passed FAST and was scored a pass.

KE-160 (CA50213): The missile passed CFTS undergoing 27 cycles (109.53 test hours). The missile has eight type I BIT failures during:

- Cycle 2, BIT 4 (7.06 test hours) - PROM CHKSUM
- Cycle 3, BIT 1 (8.20 test hours) - PROM CHKSUM
- Cycle 3, BIT 3 (10.20 test hours) - PROM CHKSUM
- Cycle 4, BIT 1 (12.25 test hours) - PRF/PDI CONTROL
- Cycle 19, BIT 1 (73.10 test hours) - PROM CHKSUM
- Cycle 20, BIT 1 (77.16 test hours) - PROM CHKSUM

- Cycle 23, BIT 3 (91.33 test hours) - PROM CHKSUM and BUS SWITCH
- Cycle 24, BIT 4 (96.30 test hours) - PROM CHKSUM and BUS SWITCH

The missile had one type II BIT failure (Bit and Rebit: PROM CHKSUM) at cycle 11, BIT 3 (42.65 test hours). The missile passed FAST and was scored as a pass.

KE-155 (CA50214): The missile passed CFTS undergoing 26 cycles (105.47 test hours). The missile had three type I BIT failures (PROM CHKSUM) during:

- Cycle 2, BIT 3 (6.14 test hours)
- Cycle 2, BIT 5 (8.11 test hours)
- Cycle 7, BIT 2 (24.70 test hours)

The missile passed FAST and was scored as a pass.

KE-172 (CA50215): The missile failed CFTS when a type II BIT failure (EQ FAIL) occurred during cycle 9, BIT 5 (36.95 test hours). The missile had seven type I BIT failures before this type II failure:

- Cycle 1, BIT 4 (3.00 test hours) - PROM CHKSUM
- Cycle 2, BIT 2 (4.41 test hours) - EQ FAIL
- Cycle 2, BIT 3 (6.14 test hours) - PROM CHKSUM
- Cycle 2, BIT 5 (8.11 test hours) - PROM CHKSUM
- Cycle 3, BIT 2 (8.47 test hours) - PROM CHKSUM
- Cycle 4, BIT 1 (12.25 test hours) - PROM CHKSUM
- Cycle 8, BIT 4 (31.40 test hours) - EQ FAIL

The missile also had another type II BIT failure (Bit: EQ FAIL and Rebit: PROM CHKSUM & BUS SWITCH) at cycle 7, BIT 3 (26.42 test hours). As seen above, the missile had multiple intermittent BIT failures for 1553/Bus Switch during CFTS. FAST verified the failure mode. RAYCO's

failure analysis isolated the failure to a fractured soder joint on U5, pin 10 of the remote terminal. The remote terminal is a Printed Wiring Board (PWB) with leadless chip carriers containing up to 28 pins. The missile was scored a failure with 35.98 credited test hours.

KE-170 (CA50217): The missile passed CFTS undergoing 27 cycles (109.53 test hours). The missile had six type I BIT failures:

- Cycle 2, BIT 1 (4.14 test hours) - PROM CHKSUM
- Cycle 4, BIT 4 (15.17 test hours) - PROM CHKSUM
- Cycle 7, BIT 1 (24.42 test hours) - PROM CHKSUM
- Cycle 7, BIT 3 (26.42 test hours) - PROM CHKSUM
- Cycle 10, BIT 3 (38.59 test hours) - PROM CHKSUM
- Cycle 27, BIT 1(105.56 test hours) - PROM CHKSUM & BUS SWITCH

The missile had three type II BIT failures:

- Cycle 2, BIT 4 (7.06 test hours) - Bit: PROM CHKSUM & Rebit: EQ FAIL
- Cycle 23, BIT 1 (89.33 test hours) - Bit and Rebit: PROM CHKSUM & BUS SWITCH
- Cycle 23, BIT 2 (89.60 test hours) - Bit and Rebit: PROM CHKSUM & BUS SWITCH

The missile passed FAST and was scored as a pass.

KE-174 (CA50218): The missile passed CFTS undergoing 27 cycles (109.53 test hours). The missile had four type I BIT failures:

- Cycle 7, BIT 3 (26.42 test hours) - PROM CHKSUM & BUS SWITCH
- Cycle 12, BIT 3 (46.71 test hours) - PROM CHKSUM
- Cycle 18, BIT 3 (71.05 test hours) - PROM CHKSUM & BUS SWITCH
- Cycle 20, BIT 3 (79.16 test hours) - PROM CHKSUM

The missile passed FAST and was scored a pass.

KE-183 (CA50219): The missile failed CFTS when a type II BIT failure (Bit & Rebit: AGC INIT) occurred at cycle 6, BIT 4 (23.28 test hours). A type I BIT failure (PROM CHKSUM & BUS SWITCH) occurred before at cycle 3, BIT 5 (12.17 test hours). The missile failed CFTS for multiple radar faults. The fault was verified by FAST at the factory. However, the fault was intermittent and usually showed up while the missile was exposed to vibration. The guidance section failed BIT under low level vibration, and passed at static conditions. RAYCO believed the fault to be in the Frequency Reference Unit (FRU). The missile was scored a failure with 22.83 credited test hours.

KE-185 (CA50220): The missile passed CFTS undergoing 23 cycles (93.30 test hours). The missile had five type I BIT failures:

- Cycle 15, BIT 4 (59.79 test hours) - PROM CHKSUM & BUS SWITCH
- Cycle 18, BIT 1 (69.05 test hours) - PROM CHKSUM & BUS SWITCH
- Cycle 19, BIT 3 (75.10 test hours) - PROM CHKSUM
- Cycle 19, BIT 5 (77.08 test hours) - PROM CHKSUM & BUS SWITCH
- Cycle 20, BIT 1 (77.16 test hours) - PROM CHKSUM & BUS SWITCH

The missile had two type II BIT failures:

- Cycle 17, BIT 4 (67.91 test hours) - Bit & Rebit: PROM CHKSUM
- Cycle 20, BIT 3 (79.16 test hours) - Bit & Rebit: PROM CHKSUM & BUS SWITCH

The missile experienced a FAST failure which was not BIT detectable. The FAST was performed three times with similar results each time. The missiles had false detection's (spurious frequencies) in multiple processor cells for each failed radar mode. Engineering personnel conclude the Medium Pulse Repetition Frequency (MPRF) Bird was a parametric fault and considered a non-relevant failure. The missile was rescored a pass.

KE-189 (CA50272): The missile passed CFTS undergoing 27 cycles (109.53 test hours). However, the missile had seven type I BIT failures:

- Cycle 10, BIT 3 (38.59 test hours) - PROM CHKSUM & NO INRT 8
- Cycle 11, BIT 3 (42.65 test hours) - PROM CHKSUM & NO INRT 8
- Cycle 14, BIT 4 (55.74 test hours) - PROM CHKSUM & NO INRT 8
- Cycle 19, BIT 5 (77.08 test hours) - PROM CHKSUM & NO INRT 8
- Cycle 20, BIT 5 (81.13 test hours) - PROM CHKSUM & BUS SWITCH
- Cycle 21, BIT 2 (81.49 test hours) - PROM CHKSUM & BUS SWITCH

and two type II failures:

- Cycle 17, BIT 2 (65.26 test hours) - Bit & Rebit: PROM CHKSUM & NO INRT 8
- Cycle 22, BIT 2 (85.55 test hours) - Bit & Rebit: PROM CHKSUM & NO INRT 8

The missile passed FAST and was scored a pass.

KE-192 (CA50273): The missile passed CFTS undergoing 23 cycles (93.30 test hours). However, the missile had three type I BIT failures:

- Cycle 16, BIT 5 (64.91 test hours) - PROM CHKSUM & BUS SWITCH
- Cycle 20, BIT 3 (79.16 test hours) - PROM CHKSUM & BUS SWITCH
- Cycle 22, BIT 2 (85.55 test hours) - PROM CHKSUM & BUS SWITCH

and two type II failure:

- Cycle 17, BIT 2 (65.26 test hours) - Bit and Rebit: PROM CHKSUM & BUS SWITCH
- Cycle 21, BIT 2 (81.49 test hours) - Bit and Rebit: PROM CHKSUM & BUS SWITCH

The missile experienced a FAST failure which was not BIT detectable. FAST analysis at NAWCW-PNS indicated that the launch sequencer was faulty. RAYCO isolated the failure to a fractured pin on the launch sequencer power switching hybrid, pin number 1. The missile was scored a failure with 46.65 credited test hours (half the CFTS test time).

Summary: 10 missiles tested and 10 data points

- 3 failures (KE-172, 183, 192)
- 7 passes (KE-157,160, 155, 170, 174, 185, 189)

Recommendation:

Based on the severity of the PROM CHKSUM type failures the following missile should be recorded as failures:

- KE-160: type II BIT failure (Bit and Rebit: PROM CHKSUM) at cycle 11, BIT 3 (42.65 test hours).
- KE-170: type II BIT failure (Bit and Rebit: PROM CHKSUM & BUS SWITCH) at cycle 23, BIT 1 (89.33 test hours) .
- KE-185: type II BIT failure (Bit & Rebit: PROM CHKSUM) at Cycle 17, BIT 4 (67.91 test hours).
- KE-189: type II failures (Bit & Rebit: PROM CHKSUM & NO INRT 8) at Cycle 17, BIT 2 (65.26 test hours).

Suggest KE-192's failure point be recorded as type II failure (Bit and Rebit: PROM CHKSUM & BUS SWITCH) at Cycle 17, BIT 2 (65.26 test hours).

Raytheon Lot 3, Sublot 1:

KF-036 (CA50351): The missile failed CFTS when a type II BIT failure (Bit & Rebit: Actuator Failure) at Cycle 17, BIT 4 (67.91 test hours). The missile passed FAST, but failed

CFTS BITs during several cycles and failed ambient BIT in the chamber after two different cycles. It was considered an intermittent environmental failure. The missile was scored a failure with 67.45 credited test hours.

KF-040 (CA50352): The missile passed CFTS undergoing 28 cycles (113.59 test hours). No BIT failures were detected. The missile failed the first FAST due to a test equipment problem and then passed a second FAST. The missile was scored a pass.

KF-056 (CA50353): The missile failed CFTS when a type II BIT failure (Bit & Rebit: Seeker Rate Mode & Seeker Position Mode) was detected during Cycle 1, BIT 1 (0.08 test hours). The missile failed a total of 37 out of 53 BITs . The missile failed FAST. The failure was confirmed by a FAST at NAWCWPNS and at RAYCO in Lowell, Massachusetts. The failure mode was isolated to a hybrid on the Digital Interface Card of the antenna chassis. A digital interface hybrid shortened internally from a loose extraneous wire. The missile was scored a failure with 0.04 credited test hours.

KF-080 (CA50354): The missile failed CFTS when a type II BIT failure (Bit & Rebit: Eq Failure, could not upload umbilical simulator) was detected in Cycle 21, BIT 3 (83.22 test hours). The missile tripped the SRTS 400 Hz circuit breakers and failed for 400 Hz overload. The missile was not subjected to FAST; doing so would have increased the risk of further damaging the missile and possibly, the MTS. RAYCO's failure analysis on the missile began with missile resistance measurements. They found that the 400 Hz return line was open, the -150 volt line in the guidance section was shorted to ground, and the -135 volt line on the filter rectifier was open. RAYCO deshelled the guidance section and discovered that an electrical overload occurred along the ground plane of the guidance section backplane. Also, the In-Rush Current Limiter (LCL) hybrid's interconnect leads on the filter rectifier were electrical overloaded creating an open circuit. The root cause was likely due to metallic debris (probably a titanium particle from underneath a doubler on the aft shell) which shorted between two pins, PIN A52 (-135 volt) and PIN A54

(ground) on the backplane connector causing the backplane ground lines to fuse open. The fused material migrated and made a connection to ground at the printed wiring board mounting screw near the J4 connector. The grounding shorts caused the filter rectifier ICL hybrid to fail. The missile was scored a fail with 82.35 credited test hours.

KF-101 (CA50355): The missile passed CFTS undergoing 47 cycles (190.66 test hours). No BIT failures were detected. The missile failed the first FAST with a TDD relay hang-up in station doppler frequency signal and passed the second FAST. The missile was scored a pass.

KF-102 (CA50356): The missile passed CFTS undergoing 47 cycles (190.66 test hours). No BIT failures were detected. The missile pass FAST. The missile was scored a pass.

KF-138 (CA50357): The missile passed CFTS undergoing 34.60 cycles (140.37 credited test hours). The missile also experienced two separate actuator failures. A type I actuator failure occurred in Cycle 20, BIT 4 (80.08 test hours) and a type II actuator failure occurred in Cycle 29, BIT 4 (116.59 test hours). The missile pass FAST. The FAST was performed in tactical configuration since no telemetry-lock could be obtained. The missile was scored a pass.

KF-148 (CA50358): The missile passed CFTS undergoing 43.60 cycles (176.88 credited test hours). The missile had a type II BIT failure (Bit & Rebit: No INRT 7, Multiple Radar Failures) at Cycle 28, BIT 5 (113.587 test hours). The missile failed four of four ambient BITs after Cycle 28 and was removed from test. The missile was reinstalled and tested for 16 cycles with no additional failures. The missile failed the initial FAST and passed a second FAST. The missile was scored a pass.

KF-139 (CA50359): The missile passed CFTS undergoing 42 cycles (170.38 test hours). No BIT failures were detected. The missile passed FAST. FAST was performed in tactical mode. The missile was scored a pass.

KF-167 (CA50360): The missile passed CFTS undergoing 35 cycles (141.98 test hours). No BIT failures were detected. FAST was performed in tactical mode due to telemetry data breaks. The missile should be scored a pass.

Summary: 10 missiles were tested and 10 data points were used

- three missile failures (KF-036, 056, 080)
- seven missile failures (KF-040, 101, 102, 138, 148, 139, 167)

Recommendation:

- Record KF-138 as a failure based on type II BIT detection at Cycle 29, BIT 4 (116.59 test hours).
- Record KF-148 as a failure based on type II BIT detection at Cycle 28, BIT 5 (113.587 test hours).

Raytheon Lot 3, Sublot 2:

KF-197 (CA50501): The missile passed CFTS undergoing 39 cycles (158.21 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

KF-199 (CA50502): The missile passed CFTS undergoing 39 cycles (158.21 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

KF-223 (CA50503): The missile passed CFTS undergoing 39 cycles (158.21 test hours). The missile had a type II BIT failure (Bit & Rebit: AGC INIT, PHASE INIT, PN CODE) at Cycle 1, BIT 3 (2.08 test hours). The missile failed two FASTs for Antenna Azimuth Compensate Discriminate Ratio. The ratios measured were 0.79055 and 0.78688 degrees/degree, respectively. The upper and lower bounds for this ratio are specified as 1.300 ¿ measured value ¿ 0.800. The data was inconclusive for scoring the missile as a pass at the level of analysis available at NAWCWPNS-Point Mugu. After the missiles were returned to RAYCO, another FAST test was conducted at

Letterkenny with similar results. The missile was scored a pass (no documentation to confirm as yet).

KF-219 (CA50504): The missile passed CFTS undergoing 39 cycles (158.21 test hours). No BIT failures were detected. The missile was FASTed four times. On the first, second, and four FAST attempts, the missile had data breaks. The third FAST attempt was not finished. However; data available from FAST BIT indicates a good missile in the area where the data breaks occurred during the HV portion of FAST. The missile was scored as a pass.

KF-258 (CA50505): The missile passed CFTS undergoing 32.02 cycles (129.88 test hours). No BIT failures were detected. The missile experienced problems with the telemetry pack during attempts to complete a FAST. It failed the initial FAST for RADAR Input/Output. The missile was retested, bypassing the telemetry pack with a telemetry simulator, as a tactical missile and passed the FAST. The missile was scored as a pass.

KF-263 (CA50506): The missile passed CFTS undergoing 39 cycles (158.21 test hours). No BIT failures were detected. The missile failed the first FAST for maximum (relative average) apaccy; however, after analyzing the supporting data, it was determined that the failure was inconclusive. The missile was reFASTed and again failed due to a malfunctioning test equipment counter. The first and second FASTs were combined and the missile was scored a pass.

KF-286 (CA50507): The missile passed CFTS undergoing 39 cycles (158.21 test hours). The missile had two type II BIT failures:

- Cycle 5, BIT 4 (19.23 test hours) - Bit & Rebit: Actuator Failure
- Cycle 5, BIT 5 (20.28 test hours) - Bit & Rebit: Actuator Failure
- Cycle 6, BIT 1 (20. 38 test hours) - Bit & Rebit: Actuator Failure

The missile failed the initial FAST for actuators (pressure at low specification). The missile also failed the second FAST for -20V standard deviation control specification. Support data indicated

the failure in -20V standard deviation was telemetry pack related. Combining the results of the first and second FAST, FAST was scored as a pass and thus, the missile was scored a pass.

KF-290 (CA50508): The missile passed CFTS undergoing 32.02 cycles (129.88 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

KF-314 (CA50509): The missile passed CFTS undergoing 32.02 cycles (129.88 test hours). No BIT failures were detected. The missile failed an initial FAST for Servo Pos Linearity and a second FAST for TDD Noise measurements. Upon analyzing the data, the government determined the failures to be test station related and scored the missile as a pass.

KF-320 (CA50510): The missile passed CFTS undergoing 32.02 cycles (129.88 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

Summary: 10 missiles were tested and 10 missiles were used as data points

- 0 failures
- 10 missile were scored as passes

Recommendation:

- Record KF-223 as a failure for a type II BIT failure at Cycle 1, BIT 3 (2.08 test hours).
- Record KF-286 as a failure for a type II BIT failure at Cycle 5, BIT 4 (19.23 test hours).

Raytheon Lot 4, Sublot 1:

KG-053 (CA50751): The missile passed CFTS undergoing 37 cycles (150.10 test hours). No BIT failures were detected. The missile failed FAST for "Warhead Fire number 2" number of transitions and pulse width. A continuity check of the missile guidance arm/test connector (17J3) revealed an open pin 60 to ground on the missile. This failure mode was not detectable by BIT. The missile was scored a failure with half its CFTS hours: 18.5 cycles (75.05 credited test hours).

KG-058 (CA50752): The missile passed CFTS undergoing 37 cycles (150.10 test hours). The missile had a type I BIT failure (PROM CHECKSUM) at cycle 23, BIT 2 (89.60 test hours). The missile passed FAST and was scored a pass.

KG-074 (CA50753): The missile passed CFTS undergoing 45 cycles (182.55 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

KG-081 (CA50754): The missile passed CFTS undergoing 45 cycles (182.55 test hours). The missile had three type I BIT failures (PROM CHKSUM):

- Cycle 12, BIT 1 (44.71 test hours)
- Cycle 37, BIT 2 (146.40 test hours)
- Cycle 39, BIT 4 (157.15 test hours)

The missile failed FAST for "Warhead Fire number 2" number of transitions and pulse width. A continuity check of the missile guidance arm/test connector (17J3) revealed an open pin 60 to ground on the missile. This failure mode was not detectable by BIT. The missile was scored a failure with half its CFTS hours: 22.5 cycles (91.28 credited test hours).

KG-127 (CA50755): The missile failed CFTS when a type II failure (Bit & Rebit: AGC INIT) occurred at cycle 16, BIT 4 (63.85 test hours). The missile had two type I BIT failures before the type II BIT failure:

- Cycle 10, BIT 3 (38.59 test hours) - PROM CHKSUM
- Cycle 15, BIT 4 (59.79 test hours) - AGC INIT

Failure persisted throughout the remainder of CFTS. The missile failed FAST for multiple IF receiver, range correlator, filter processor, Inertial Reference Unit (IRU), and Frequency Reference Unit (FRU) failure which confirmed the failure seen during BITs conducted in CFTS. The failure was isolated to the FRU microwave assembly. The missile was scored a failure with 63.39 credited test hours.

KG-131 (CA50756): The missile passed CFTS undergoing 45 cycles (182.55 test hours). The missile had a type I BIT failure (PROM CHKSUM) at cycle 10, BIT 3 (38.59 test hours). The missile passed FAST and was scored a pass.

KG-149 (CA50757): The missile passed CFTS undergoing 57 cycles (231.23 test hours). No BIT failures were detected. The missile was FASTed after 26 cycles and failed for Filter Processor PDI 10 Rel 12. The missile passed a second FAST with nominal result. The failed FAST was determined to be non-relevant. The missile was rebuilt and where tested for an additional 31 cycles. The missile passed another FAST and was scored as a pass.

KG-172 (CA50758): The missile failed CFTS when a type II BIT failure (Bit: PROM CHKSUM, ACTUATOR; Rebit: PROM CHKSUM) was detected a cycle 44, BIT 1 (174.52 test hours). The missile experienced numerous Type II (PROM CHKSUM) failures. Several BITs failed for PROM CHKSUM on umbilical data, but the telemetry data showed only actuator failures. When FAST was conducted the missile passed the BIT portion of FAST, but failed FAST for multiple filter processor and seeker servo position linearity failures. RAYCO was unable to duplicate the failure mode and indicated that the failure may be due to a non-tactical harness cable. The missile was scored as a failure with 174.48 credited test hours.

KG-194 (CA50759): The missile passed CFTS undergoing 53 cycles (215.00 test hours). No BIT failures were detected. The missile failed FAST for seeker servo elevation rate linearity number 3 verify by position using FAST software version 115. The missile would pass using version 116 software limits. The missile was scored as a pass.

KG-190 (CA50760): The missile passed CFTS undergoing 53 cycles (215.00 test hours). No BIT failures were detected. The missile failed FAST for seeker servo position linearity 1. These failures were determined to be non-relevant and the missile was scored as a pass.

KG-220 (CA50761): The missile passed CFTS undergoing 64 cycles (259.63 test hours). No BIT failures were detected. The missile was FASTed after 33 cycles and passed. The missile was

rebuilt and tested for another 31 test cycles. The missile was subjected to another FAST and passed. The missile was scored as a pass.

KG-217 (CA50762): The missile passed CFTS undergoing 57 cycles (231.23 test hours). The missile had two type II BIT failures (Bit & Rebit: RDI Continuity):

- Cycle 43, BIT 3 (172.46 test hours)
- Cycle 48, BIT 3 (192.75 test hours)

The missile was FASTed after 33 cycles and passed. The missile was rebuilt and tested for another 24 test cycles. The missile was subjected to another FAST and passed. The missile was scored as a pass.

Summary: 12 missiles tested and 12 missiles used as test points

- 4 failures (KE-053,081,127,172)
- 8 passes (KE-058,074,131,149,190,194,220,217)

Recommendation:

- Record KG-053 with full CFTS time, 37 cycles (150.10 test hours), as a result of no type II BIT failures.
- Record KG-081 with full CFTS time, 45 cycles (182.55 test hours), as a result of no type II BIT failures.
- Record KG-217 as a failure based on type II failure at Cycle 43, BIT 3 (172.46 test hours).

Raytheon Lot 4, Sublot 2:

KG-237 (CA50901): The missile passed CFTS undergoing 34 cycles (137.93 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

KG-278 (CA50902): The missile passed CFTS undergoing 34 cycles (137.93 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

KG-284 (CA50903): The missile passed CFTS undergoing 34 cycles (137.93 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

KG-291 (CA50904): The missile passed CFTS undergoing 34 cycles (137.93 test hours). The missile had a type I BIT failure (RDI CONTINUITY) at cycle 1, BIT 2 (0.36 test hours) and a type II BIT failure (Bit: HPRF mode, TGT image and Rebit: AGC Init, HPRF Mode, TGT image) at cycle 7, BIT 3 (26.42 test hours). The missile passed FAST and was scored a pass.

KG-333 (CA50905): The missile failed CFTS when a type II BIT failure (Bit & Rebit: EQ FAIL) at cycle 40, BIT 3 (160.29 test hours). The missile tripped the SRTS circuit breakers for 400 Hz power both times BIT was attempted. Failure analysis isolated the failure to an overstressed In-rush Current Limiter (ICL) hybrid on the filter rectifier. The filter rectifier was removed and replaced and the missile passed a FAST. The missile was scored a failure with 159.43 credited test hours.

KG-340 (CA50906): The missile passed CFTS undergoing 50 cycles (202.83 test hours). The missile had a type II BIT failure (Bit & Rebit: TDD failure) at cycle 33, BIT 1 (129.83 test hours). The missile passed FAST and was scored as a pass.

KG-369 (CA50907): The missile passed CFTS undergoing 50 cycles (202.83 test hours). No BIT failures were detected. The missile passed FAST and was scored as a pass.

KG-378 (CA50908): The missile passed CFTS undergoing 50 cycles (202.83 test hours). The missile had a type I BIT failure (VCXO INIT) at cycle 15, BIT 4 (59.79 test hours). The missile passed FAST and was scored as a pass.

KG-387 (CA50909): The missile passed CFTS undergoing 44 cycles (178.49 test hours). The missile had a type I BIT failure (TDD FAILURE) at cycle 9, BIT 1 (32.54 test hours). The missile passed FAST and was scored a pass.

KG-417 (CA50910): The missile passed CFTS undergoing 34 cycles (137.93 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

KG-431 (CA50911): The missile passed CFTS undergoing 50 cycles (202.83 test hours). No BIT failures were detected. The missile passed FAST and was scored as a pass.

KG-433 (CA50912): The missile passed CFTS undergoing 50 cycles (202.83 test hours). No BIT failures were detected. The missile passed FAST and was scored as a pass.

Summary: 12 missiles were tested and 12 missiles were used as data points

- 1 failure (KE-333)
- 11 passes

Recommendation:

- Record KG-291 as a failure base on type II BIT failure at cycle 7, BIT 3 (26.42 test hours).
- Record KG-340 as a failure base on type II BIT failure at cycle 33, BIT 1 (129.83 test hours).

Raytheon Lot 5, Sublot 1:

KH-011 (CA51151): The missile passed CFTS undergoing 53 cycles (215.00 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

KH-019 (CA51152): The missile passed CFTS undergoing 48.07 cycles (195.00 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

KH-031 (CA51153): The missile passed CFTS undergoing 39 cycles (158.21 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

KH-033 (CA51154): The missile passed CFTS undergoing 48.07 cycles (195.00 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

KH-037 (CA51155): The missile failed CFTS when a type II BIT failure (Bit & Rebit: Seeker Position Mode, Seeker Rate Mode) was detected at cycle 13, BIT 4 (51.68 test hours). These failures were confirmed by FAST. The failure was isolated to a fractured coil lead on a torque amplifier

board in the antenna. The lead fractured because staking epoxy on this coil was inadvertently missed by the operator. The missile was scored a failure with 51.22 credited test hours.

KH-039 (CA51156): The missile failed CFTS when a type II BIT failure (Bit: ADP Stack, Mult Fails & Rebit: IRU BIT STATUS) on cycle 11, BIT 1 (40.65 test hours). This failure was not confirmed by FAST. Type II BIT failures were recorded beginning with cycle 11, BIT 1; cycle 19, BIT 1; cycle 26, BIT 5; cycle 34, BIT 5; and cycle 39, BIT 5. With exception of the fault during cycle 39, BIT 5, each fault occurred when the missile's internal temperature, as read from the telemetry T1/T2 outputs, was greater than 37 degrees Celsius. The contractor established the fault in the guidance section and was continuing the investigation. The missile was scored a failure with 40.61 credited test hours.

KH-049 (CA51157): The missile passed CFTS undergoing 39 cycles (158.21 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

KH-072 (CA51158): The missile passed CFTS undergoing 39 cycles (158.21 test hours). The missile had two type I BIT failures (RDI Continuity) at cycle 5, BIT 4 (19.23 test hours) and cycle 6, BIT 4 (23.28 test hours). The missile passed FAST and was scored a pass.

KH-068 (CA51159): The missile passed CFTS undergoing 34.07 cycles (138.21 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

KH-117 (CA51160): The missile passed CFTS undergoing 34.07 cycles (138.21 test hours). No BIT failures were detected. The missile failed FAST at Letterkenney for Azimuth RF Boresight. Engineering personnel concluded that this fault was not a reliability failure and deemed a non-relevant PRAT failure. The missile was scored a pass.

KH-113 (CA51161): The missile passed CFTS undergoing 53 cycles (215.00 test hours). The missile had a type I BIT failure (VCXO INIT) at cycle 53, BIT 3 (213.03 test hours). The missile passed FAST and was scored a pass.

KH-123 (CA51162): The missile passed CFTS undergoing 54 cycles (219.06 test hours). The missile had a type I BIT failure at cycle 23, BIT 1 (89.33 test hours). The missile passed FAST and was scored a pass.

Summary: 12 missile tested and 12 data points used

- 2 failures (KH-037, 039)
- 10 passes (KH-011, 019, 031, 033, 049, 072, 068, 117, 113, 123)

Recommendation: - None

Raytheon Lot 5, Sublot 2: Have only handwritten notes as background material. No FAST data.

KH-133 (CA51301): The missile passed CFTS undergoing 50 cycles (202.83 test hours). No BIT failures were detected. The missile was scored a pass.

KH-136 (CA51302): The missile failed CFTS when a type II BIT failure (Bit & Rebit: AGC INIT) was detected at cycle 30, BIT 3 (119.73 test hours). The missile was scored a failure with 118.86 credited test hours.

KH-147 (CA51303): The missile failed CFTS when a type II BIT failure (Bit & Rebit: IRU TAG) was detected at cycle 14, BIT 4 (55.74 test hours). The missile was scored a failure with 55.28 credited test hours.

KH-156 (CA51304): The missile passed CFTS undergoing 65 cycles (263.68 test hours). The missile had a type I BIT failure (RDI Continuity) at cycle 14, BIT 2 (53.09 test hours). The missile was scored a pass.

KH-166 (CA51305): The missile passed CFTS undergoing 22 cycles (89.25 test hours). The missile had a type I BIT failure (Actuator) at cycle 22, BIT 3 (87.27 test hours). The missile was scored a pass.

KH-174 (CA51306): The missile passed CFTS undergoing 65 cycles (263.68 test hours). The missile had a type I BIT failure (RDI Continuity) at cycle 49, BIT 1 (194.80 test hours). The missile was scored a pass.

KH-188 (CA51307): The missile failed CFTS when a type II BIT failure (Bit & Rebit: TDD) was detected at cycle 45, BIT 5 (182.55 test hours). The missile was scored a failure with 182.02 credited test hours.

KH-202 (CA51308): The missile failed CFTS when a type II BIT failure (Bit & Rebit: Actuator) was detected at cycle 1, BIT 1 (0.808 test hours). The missile was scored a failure with 0.04 credited test hours.

KH-196 (CA51309): The missile passed CFTS undergoing 77 cycles (312.36 test hours). The missile had a type I BIT failure (VCXO) at cycle 48, BIT 4 (193.66 test hours). The missile was scored a pass.

KH-245 (CA51310): The missile passed CFTS undergoing 53 cycles (215.00 test hours). No BIT failures were detected. The missile was scored a pass.

KH-251 (CA51311): The missile failed the incoming BIT and was not scored.

KH-261 (CA51312): The missile passed CFTS undergoing 53 cycles (215.00 test hours). No BIT failures were detected. The missile was scored a pass.

Summary: 12 missiles tested and 11 data points used

- one missile (KH-251) failed incoming BIT
- four missiles (KH-136,147,188,202) failed
- seven missiles (KH-133,156,166,174,196,245,261)

Recommendation: - None.

Raytheon Lot 6, Sublot 1:

KI-021 (CA51451): The missile failed CFTS when a type II BIT failure (Bit & Rebit: ACTUATOR) was detected at cycle 1, BIT 5 (4.06 test hours). The missile intermittently failed BITs 1, 4, and 5 through 14 cycles. Failure mode was actuator scan ok 2 and 4. The missile passed FAST at Letterkenney. The missile was scored a failure with 3.53 credited test hours.

KI-024 (CA51452): The missile failed CFTS when a type II BIT failure (Bit: AGC INIT, HPRF MODE; Rebit: PHASE INIT, HPRF MODE) was detected at cycle 1, BIT 2 (0.36 test hours). The missile consistently failed BITs 1 through 4 until removed after eight cycles. The missile failures, HPRF MODE and PHASE INIT were confirmed by FAST at Letterkenney. The missile was scored a failure with 0.22 credited test hours.

KI-070 (CA51453): The missile passed CFTS undergoing 97 cycles (393.50 test hours). The missile had a type I BIT failure (VCXO INIT) at cycle 58, BIT 5 (239.34 test hours). The missile passed FAST and was scored a pass.

KI-081 (CA51454): The missile failed CFTS when a type II BIT failure (Bit & Rebit: PROM CHKSUM, MULT FAILS) was detected at cycle 5, BIT 5 (20.83 test hours). The missile had a total of six type II PROM CHKSUM failures in 14 cycles. The failure was confirmed by FAST and the missile was scored a failure with 19.76 credited test hours.

KI-096 (CA51455): The missile passed CFTS undergoing 89 cycles (361.04 test hours). The missile had a type II BIT failure (Bit & Rebit: AGC COMMAND) at cycle 36, BIT 3 (144.07 test hours) at cycle 70, BIT 3 (281.99 test hours). The missile passed FAST and was scored a pass.

KI-138 (CA51456): The missile passed CFTS undergoing 86 cycles (348.87 test hours). No BIT failures were detected. The missile passed FAST and was scored a pass.

KI-164 (CA51457): The missile passed CFTS undergoing 96 cycles (389.44 test hours). The missile had a type I BIT failure (VCXO INIT) at cycle 38, BIT 5 (154.15 test hours) and at cycle 42, BIT 5 (170.38 test hours). The missile passed FAST and was scored a pass.

KI-190 (CA51458): The missile passed CFTS undergoing 78.64 cycles (319.03 test hours). The missile had eleven type I BIT failures:

- Cycle 17, BIT 4 (67.91 test hours): MPRF MODE
- Cycle 33, BIT 1 (129.90 test hours): MPRF MODE
- Cycle 40, BIT 1 (158.29 test hours): EQ FAIL
- Cycle 40, BIT 3 (160.29 test hours): PROM CHKSUM, MULT FAILS
- Cycle 65, BIT 4 (262.63 test hours): EQ FAIL
- Cycle 67, BIT 1 (267.82 test hours): EQ FAIL
- Cycle 68, BIT 3 (273.88 test hours): PROM CHKSUM, MULT FAILS
- Cycle 69, BIT 1 (275.94 test hours): PROM CHKSUM, MULT FAILS
- Cycle 69, BIT 2 (276.21 test hours): PROM CHKSUM, MULT FAILS
- Cycle 77, BIT 1 (308.39 test hours): PROM CHKSUM, MULT FAILS
- Cycle 78, BIT 3 (314.45 test hours): EQ FAIL

The missile had the following type II BIT failures:

- Cycle 64, BIT 4 (258.57 test hours): Bit & Rebit - EQ FAIL
- Cycle 74, BIT 4 (299.14 test hours): Bit & Rebit - EQ FAIL
- Cycle 79, BIT 1 (316.50 test hours): Bit & Rebit - EQ FAIL
- Cycle 79, BIT 2 (316.78 test hours): Bit & Rebit - EQ FAIL
- Cycle 79, BIT 3 (318.50 test hours): Bit & Rebit - DAGC INIT

The test was halted for investigation of umbilical cable damage. The failures were potentially induced by test equipment. The missile was scored a pass.

KI-209 (CA51459): The missile failed CFTS when a type II BIT failure:

- Bit: DAGC INIT, AGC INIT, PHASE INIT, DAGC CMD, AGC CMD, HPRF MODE, PN CODE, TGT IMAGE
- Rebit: DAGC INIT, PHASE INIT, DAGC CMD, HPRF MODE, PN CODE, TGT IMAGE

  was detected at cycle 28, BIT 5 (113.59 test hours).

The missile failed for multiple radar faults. The missile passed FAST and was scored a failure with 113.06 credited test hours.

KI-217 (CA51460): The missile passed CFTS undergoing 102.64 cycles (416.39 test hours). No BIT failures occurred. The missile passed FAST and was scored a pass.

KI-241 (CA51461): The missile failed CFTS when a type II BIT failure (Bit & Rebit: TDD) was detected at cycle 3, BIT 4 (11.11 test hours). The missile failed intermittently at guidance section temperatures between 12 and 24 degrees Celsius as temperature ramped up from cold soak. The failed parameter were TDD high and low clock. The missile passed FAST. The missile was scored a failure with 10.66 credited test hours.

KI-257 (CA51462): The missile passed CFTS undergoing 60.64 cycles (246.01 test hours). The missile had three type I BIT failures;

- Cycle 43.64, BIT 1 (177.13 test hours) - ACTUATOR
- Cycle 44.64, BIT 4 (184.10 test hours) - ACTUATOR
- Cycle 45.64, BIT 4 (188.16 test hours) - ACTUATOR

The missile had seventeen type II BIT failures with the first one occurring at Cycle 40.64, BIT 5 (164.88 test hours) - Bit & Rebit: ACTUATOR. The missile failed FAST, but was scored a pass.

Summary: 12 missiles were tested and 12 missiles were used as data points

- 5 missiles failed (KI-021,024,081,209,241)
- 7 missiles passed (KI-070,096,138,164,190,217,257)

Recommendation:

- Record KI-096 as a failure based on type II BIT failure at cycle 36, BIT 3 (144.07 test hours).
- Record KI-190 as a failure based on type II BIT failure at cycle 64, BIT 4 (258.57 test hours).
- Record KI-257 as a failure based on type II BIT failure at cycle 40.64, BIT 5 (164.88 test hours).

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
KE-157	R2-3	0	C27	109.53	141.03	0	C27	109.53	141.03
KE-160	R2-3	0	C27	109.53	141.03	1	C11,B3	41.79	53.70
KE-155	R2-3	0	C26	105.47	135.81	0	C26	105.47	135.81
KE-172	R2-3	1	C9,B5	35.98	46.04	1	C9,B5	35.98	46.04
KE-170	R2-3	0	C27	109.53	141.03	1	C23,B1	89.29	114.96
KE-174	R2-3	0	C27	109.53	141.03	0	C27	109.53	141.03
KE-183	R2-3	1	C6,B4	22.83	29.03	1	C6,B4	22.83	29.03
KE-185	R2-3	0	C23	93.30	120.14	1	C17,B4	67.45	86.49
KE-189	R2-3	0	C27	109.53	141.03	1	C17,B2	65.13	83.92
KE-192	R2-3	1	C11.5	46.65	60.07	1	C17,B2	65.13	83.92
		3		851.89	1096.23	7		712.12	915.92
			$\hat{\lambda}$	283.96	365.41		$\hat{\lambda}$	101.73	130.85

Table 18. Raytheon Lot 2 Sublot 3 Sampling Data

Below are the sampling data sets used for the Raytheon PRAT analysis[2]. The abbreviations are as follows:

- *FAST* - results based on FAST criteria (0 - right censored, 1 - uncensored).
- *BIT* - results based on BIT criteria (0 - right censored, 1 - uncensored).
- *C,B* - Cycle, BIT (Cycle and bit the missile failed a bit on. For example, C6,B3 indicates that the missile was considered to have failed on the third bit of the sixth cycle. If no bit is indicated, then the missile passed PRAT. For example, C37 indicates that the missile passed PRAT undergoing 37 cycles.).
- *Vib Hrs* - Vibration Only test time.
- *Total Hrs* - Total test time.
- $\hat{\lambda}$ - Maximum likehood estimate of the parameter of the Exponential distribution.

[2]Failure times are recorded at the midpoint between BIT checks

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
KF-036	R3-1	1	C17,B4	67.45	86.49	1	C17,B4	67.45	86.49
KF-040	R3-1	0	C28	113.59	146.25	0	C28	113.59	146.25
KF-056	R3-1	1	C1,B1	0.04	0.04	1	C1,B1	0.04	0.04
KF-080	R3-1	1	C21,B3	82.35	105.94	1	C21,B3	82.35	105.94
KF-101	R3-1	0	C47	190.66	245.50	0	C47	190.66	245.50
KF-102	R3-1	0	C47	190.66	245.50	0	C47	190.66	245.50
KF-138	R3-1	0	C34.60	140.36	180.73	1	C29,B4	116.13	149.17
KF-148	R3-1	0	C43.60	176.87	227.74	1	C28,B5	113.06	145.28
KF-139	R3-1	0	C42	170.38	219.38	0	C42	170.38	219.38
KF-167	R3-1	0	C35	141.98	182.82	0	C35	141.98	182.82
		3		1274.35	1640.38	5		1186.31	1526.36
			$\hat{\lambda}$	424.78	546.79		$\hat{\lambda}$	237.26	305.27

Table 19. Raytheon Lot 3 Sublot 1 Sampling Data

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
KF-197	R3-2	0	C39	158.21	203.71	0	C39	158.21	203.71
KF-199	R3-2	0	C39	158.21	203.71	0	C39	158.21	203.71
KF-223	R3-2	0	C39	158.21	203.71	1	C1,B3	1.22	1.47
KF-219	R3-2	0	C39	158.21	203.71	0	C39	158.21	203.71
KF-258	R3-2	0	C32.02	129.89	167.25	0	C32.02	129.89	167.25
KF-263	R3-2	0	C39	158.21	203.71	0	C39	158.21	203.71
KF-286	R3-2	0	C39	158.21	203.71	1	C5,B4	18.77	23.81
KF-290	R3-2	0	C32.02	129.89	167.25	0	C32.02	129.89	167.25
KF-314	R3-2	0	C32.02	129.89	167.25	0	C32.02	129.89	167.25
KF-320	R3-2	0	C32.02	129.89	167.25	0	C32.02	129.89	167.25
		0		1468.84	1891.26	2		1172.41	1509.12
			$\hat{\lambda}$	1468.84	1891.26		$\hat{\lambda}$	586.20	754.56

Table 20. Raytheon Lot 3 Sublot 2 Sampling Data

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
KG-053	R4-1	1	C18.5	75.05	96.63	0	C37	150.10	193.26
KG-058	R4-1	0	C37	150.10	193.26	0	C37	150.10	193.26
KG-074	R4-1	0	C45	182.55	235.05	0	C45	182.55	235.05
KG-081	R4-1	1	C22.5	91.28	117.53	0	C45	182.55	235.05
KG-127	R4-1	1	C16,B4	63.39	81.27	1	C16,B4	63.39	81.27
KG-131	R4-1	0	C45	182.55	235.05	0	C45	182.55	235.05
KG-149	R4-1	0	C57	231.23	297.73	0	C57	231.23	297.73
KG-172	R4-1	1	C44,B1	174.48	224.65	1	C44,B1	174.48	224.65
KG-194	R4-1	0	C53	215.00	276.84	0	C53	215.00	276.84
KG-190	R4-1	0	C53	215.00	276.84	0	C53	215.00	276.84
KG-220	R4-1	0	C64	259.63	334.29	0	C64	259.63	334.29
KG-217	R4-1	0	C57	231.23	297.73	1	C43,B3	171.60	220.85
		4		2071.48	2666.86	3		2178.18	2804.14
			$\hat{\lambda}$	517.87	666.71		$\hat{\lambda}$	726.06	934.71

Table 21. Raytheon Lot 4 Sublot 1 Sampling Data

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
KG-237	R4-2	0	C34	137.93	177.59	0	C34	137.93	177.59
KG-278	R4-2	0	C34	137.93	177.59	0	C34	137.93	177.59
KG-284	R4-2	0	C34	137.93	177.59	0	C34	137.93	177.59
KG-291	R4-2	0	C34	137.93	177.59	1	C7,B3	25.56	32.81
KG-333	R4-2	1	C40,B3	159.43	205.18	1	C40,B3	159.43	205.18
KG-340	R4-2	0	C50	202.83	261.17	1	C33,B1	129.86	167.19
KG-369	R4-2	0	C50	202.83	261.17	0	C50	202.83	261.17
KG-378	R4-2	0	C50	202.83	261.17	0	C50	202.83	261.17
KG-387	R4-2	0	C44	178.49	229.83	0	C44	178.49	229.83
KG-417	R4-2	0	C34	137.93	177.59	0	C34	137.93	177.59
KG-431	R4-2	0	C50	202.83	261.17	0	C50	202.83	261.17
KG-433	R4-2	0	C50	202.83	261.17	0	C50	202.83	261.17
		1		2041.72	2628.81	3		1856.38	2390.05
			$\hat{\lambda}$	2041.72	2628.81		$\hat{\lambda}$	618.79	796.68

Table 22. Raytheon Lot 4 Sublot 2 Sampling Data

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
KH-11	R5-1	0	C53	215.00	276.84	0	C53	215.00	276.84
KH-19	R5-1	0	C48.07	195.00	251.09	0	C48.07	195.00	251.09
KH-31	R5-1	0	C39	158.21	203.71	0	C39	158.21	203.71
KH-33	R5-1	0	C48.07	195.00	251.09	0	C48.07	195.00	251.09
KH-37	R5-1	1	C13,B4	51.22	65.60	1	C13,B4	51.22	65.60
KH-39	R5-1	1	C11,B1	40.61	52.28	1	C11,B1	40.61	52.28
KH-49	R5-1	0	C39	158.21	203.71	0	C39	158.21	203.71
KH-72	R5-1	0	C39	158.21	203.71	0	C39	158.21	203.71
KH-68	R5-1	0	C34.07	138.21	177.96	0	C34.07	138.21	177.96
KH-117	R5-1	0	C34.07	138.21	177.96	0	C34.07	138.21	177.96
KH-113	R5-1	0	C53	215.00	276.84	0	C53	215.00	276.84
KH-123	R5-1	0	C54	219.06	282.06	0	C54	219.06	282.06
		2		1881.96	2422.82	2		1881.96	2422.82
			$\hat{\lambda}$	940.98	1211.41		$\hat{\lambda}$	940.98	1211.41

Table 23. Raytheon Lot 5 Sublot 1 Sampling Data

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
KH-133	R5-2	0	C50	202.83	261.17	0	C50	202.83	261.17
KH-136	R5-2	1	C30,B3	118.86	152.95	1	C30,B3	118.86	152.95
KH-147	R5-2	1	C14,B4	55.28	70.82	1	C14,B4	55.28	70.82
KH-156	R5-2	0	C65	263.68	339.52	0	C65	263.68	339.52
KH-166	R5-2	0	C22	89.25	114.91	0	C22	89.25	114.91
KH-174	R5-2	0	C65	263.68	339.52	0	C65	263.68	339.52
KH-188	R5-2	1	C45,B5	182.02	234.08	1	C45,B5	182.02	234.08
KH-202	R5-2	1	C1,B1	0.04	0.04	1	C1,B1	0.04	0.04
KH-196	R5-2	0	C77	312.36	402.20	0	C77	312.36	402.20
KH-245	R5-2	0	C53	215.00	276.84	0	C53	215.00	276.84
KH-251	R5-2	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
KH-261	R5-2	0	C53	215.00	276.84	0	C53	215.00	276.84
		4		1918.02	2468.87	4		1918.02	2468.87
			$\hat{\lambda}$	479.51	617.22		$\hat{\lambda}$	479.51	617.22

Table 24. Raytheon Lot 5 Sublot 2 Sampling Data

MISSILE	Lot #	FAST	C,B	Vib Hrs	Total Hrs	BIT	C,B	Vib Hrs	Total Hrs
KI-021	R6-1	1	C1,B5	3.53	4.25	1	C1,B5	3.53	4.25
KI-024	R6-1	1	C1,B2	0.22	0.35	1	C1,B2	0.22	0.35
KI-070	R6-1	0	C97	393.50	506.66	0	C97	393.50	506.66
KI-081	R6-1	1	C5,B5	19.76	25.14	1	C5,B5	19.76	25.14
KI-096	R6-1	0	C89	361.04	464.88	1	C36,B3	143.20	184.29
KI-138	R6-1	0	C86	348.87	449.21	0	C86	348.87	449.21
KI-164	R6-1	0	C96	389.44	501.44	0	C96	389.44	501.44
KI-190	R6-1	0	C78.64	319.02	410.76	1	C64,B4	258.11	331.99
KI-209	R6-1	1	C28,B5	113.06	145.28	1	C28,B5	113.06	145.28
KI-217	R6-1	0	C102.64	416.38	536.12	0	C102.64	416.38	536.12
KI-241	R6-1	1	C3,B4	10.66	13.36	1	C3,B4	10.66	13.36
KI-257	R6-1	0	C40.64	246.00	316.74	1	C40.64,B5	164.33	211.30
		5		2621.46	3374.20	8		2261.05	2909.39
			$\hat{\lambda}$	524.29	674.84		$\hat{\lambda}$	282.63	363.67

Table 25. Raytheon Lot 6 Sublot 1 Sampling Data

Failure Criteria	# of Failures	Vib Hrs	Total Hrs
FAST	22	14129.72	18189.43
	$\hat{\lambda}$	642.26	826.79
BIT	34	13166.41	16946.67
	$\hat{\lambda}$	387.25	498.43

Table 26. Raytheon Sampling Data Sets

Serial Number	Censoring	CC HOURS	LOT #
CA00509	0	7.0	3
CA00518	0	67.8	3
CA00527	0	334.3	3
CA00533	1	101.0	3
CA00535	1	82.9	3
CA00536	0	257.2	3
CA00541	0	912.5	3
CA00542	0	304.0	3
CA00543	0	498.0	3
CA00544	0	469.8	3
CA00558	1	366.0	3
CA00559	0	800.6	3
CA00560	0	872.1	3
CA00562	0	577.9	3
CA00565	0	452.3	3
CA00566	0	108.1	3
CA00567	0	922.4	3
CA00568	0	424.7	3
CA00569	0	782.3	3
CA00570	1	221.2	3
CA00571	0	131.7	3
CA00572	0	750.8	3
CA00573	0	881.1	3
CA00574	1	88.0	3
CA00576	0	387.5	3
CA00578	0	501.2	3
CA00583	0	342.2	3
CA00614	1	346.7	3
	7	11998.3	1714.0

Table 27. Italy Sampling Data Set

B.3 Hughes Missile System Company Operational Flight Sampling Data

This section details the data sets used to calculate the HMSC Operational Flight survival functions. Captive-Carry Lifelengths are as of Oct, 1994. The following abbrevations are used:

- *Censoring* - 0 right censored, 1 uncensored

- *CC Hours* - Captive-Carry hours

Serial Number	Censoring	CC HOURS	LOT #
CA00361	0	801.4	3
CA00362	1	262.3	3
CA00363	0	1216	3
CA00365	1	111.5	3
CA00366	0	225.1	3
CA00367	0	1277.8	3
CA00368	1	582.7	3
CA00369	0	923.6	3
CA00370	0	162.3	3
CA00371	0	1017.7	3
CA00373	0	169.6	3
CA00374	1	660	3
CA00375	0	346	3
CA00376	1	111.8	3
CA00377	0	821.5	3
CA00378	0	603.6	3
CA00379	1	260.4	3
CA00380	1	25.6	3
CA00384	0	779.6	3
CA00418	0	1252.6	3
CA00419	0	869	3
CA00420	1	846.7	3
CA00425	1	87.9	3
CA00428	0	659.3	3
CA00432	0	1131.6	3
CA00433	0	1095.8	3
CA00434	1	426.6	3
CA00435	1	698	3
CA00436	0	1074.3	3
CA00437	0	883.9	3
CA00438	0	1141.3	3
CA00440	0	1034.1	3
CA00441	1	57.9	3
CA00442	0	178.3	3
CA00443	0	1039.4	3
CA00445	0	336.8	3
CA00447	0	721.7	3
CA00449	1	20.9	3
CA00453	1	99.2	3
CA00456	1	555.1	3
CA00457	1	181.5	3
CA00458	0	227.9	3
S/N UNK	0	89.2	3

Table 28. Saudi Sampling Data Set

Serial Number	Censoring	CC HOURS	LOT #
CA00381	0	104.5	3
CA00382	1	293.9	3
CA00383	0	1020.8	3
CA00421	0	598.2	3
CA00422	1	496.5	3
CA00423	1	444.0	3
CA00424	0	765.9	3
CA00699	0	779.2	3
CA00709	0	708.1	3
CA00723	1	409.8	3
CA00724	1	614.8	3
CA00725	0	840.2	3
CA00726	0	746.6	3
CA00779	0	855.4	3
CA00780	0	1074.0	3
CA00524	1	650.7	3
CA00630	1	374.7	3
CA00633	0	968.3	3
CA00639	0	851.9	3
CA00660	0	1133.0	3
CA00675	0	69.6	3
CA00676	0	949.2	3
CA00677	1	104.4	3
CA00678	0	991.1	3
CA00683	0	36.9	3
CA00684	0	82.7	3
CA00685	1	178.7	3
CA00686	0	1078.5	3
CA00689	0	1079.9	3
CA00693	1	772.3	3
CA00402	1	813.5	3
CA00694	1	531.1	3
CA01478	1	120.7	5
CA01479	0	462.9	5
CA01480	0	363.9	5
CA01481	0	442.6	5
CA01482	0	396.5	5
CA01483	0	499.3	5
CA01484	1	384.3	5
CA01485	0	561.9	5
CA01486	0	471.5	5
CA01487	0	396.6	5

Table 29. Saudi Sampling Data Set - Continued #1

Serial Number	Censoring	CC HOURS	LOT #
CA01488	0	599	5
CA01489	1	463.5	5
CA01494	0	690.0	5
CA01496	0	629.6	5
CA01497	0	620.4	5
CA01498	0	603.3	5
CA01499	0	355.9	5
CA01500	0	457.9	5
CA01501	0	531.1	5
CA01502	0	459.3	5
CA01530	0	530.2	5
CA01531	0	267.5	5
CA01532	0	281.5	5
CA01533	0	421.2	5
CA01534	0	336.2	5
CA01535	0	587.9	5
CA01536	0	607.8	5
CA01537	0	652.9	5
CA01538	1	49.6	5
CA01539	0	392.8	5
CA01540	0	317.1	5
CA01541	0	193.9	5
CA01542	0	556.1	5
CA01543	0	416.6	5
CA01544	0	384.8	5
CA01545	0	424.3	5
CA01726	0	245.2	5
CA01734	0	229.5	5
CA01741	0	245.2	5
CA01745	0	245.2	5
	32	62381.6	1949.4

Table 30. Saudi Sampling Data Set - Continued #2

Serial Number	Censoring	CC HOURS	LOT #
CA00203	1	1327.7	2
CA00205	0	1201.9	2
CA00207	0	1153.9	2
CA00208	0	1344.5	2
CA00209	1	1115.6	2
CA00210	1	6.0	2
CA00216	0	1448.6	2
CA00217	0	1007.6	2
CA00219	0	1523.5	2
CA00220	0	1461.0	2
CA00223	0	40.0	2
CA00226	1	466	2
CA00232	1	34.5	2
CA00233	0	640.0	2
CA00235	0	1305.2	2
CA00236	1	1248.4	2
CA00237	0	1418.7	2
CA00240	0	877.5	2
CA00242	0	1412.3	2
CA00243	0	931.1	2
CA00245	1	40.0	2
CA00246	0	1440	2
CA00329	1	684.4	3
CA00330	0	993.4	3
CA00332	0	1422.2	3
CA00333	0	1496.0	3
CA00335	0	1432.8	3
CA00336	0	1469.3	3
CA00337	0	1441.7	3
CA00338	0	1470.0	3
CA00339	1	22.8	3
CA00340	0	1147.3	3
CA00343	0	844.5	3
CA00344	1	464.7	3
CA00345	1	582.2	3
CA00346	1	305.6	3
CA00347	1	360.5	3
CA00201	0	318.2	2
CA00211	0	1066.8	2
CA00213	1	1057.0	2
CA00214	0	1008.0	2
CA00225	1	66.5	2
CA00257	1	818.3	2
CA00277	0	250.7	2
CA00392	0	926.4	3
CA00393	0	997.4	3
CA00394	0	1026.6	3
CA00395	0	410.0	3

Table 31. Turkey Sampling Data Set

Serial Number	Censoring	CC HOURS	LOT #
CA00396	1	802.8	3
CA00397	1	836.7	3
CA00399	0	1065.6	3
CA00412	1	368.8	3
CA00413	0	917.7	3
CA00414	0	893.0	3
CA00415	1	823.9	3
CA00416	0	927.9	3
CA00417	0	916.1	3
CA00448	0	905.6	3
CA00196	0	430.7	2
CA00218	1	96.2	2
CA00228	1	162.7	2
CA00247	0	532.8	2
CA00266	1	152.9	2
CA00348	0	373.3	3
CA00349	0	391.7	3
CA00350	0	349.5	3
CA00351	0	508.2	3
CA00372	0	550.7	3
CA00386	0	449.7	3
CA00391	1	405.5	3
CA00400	0	505.7	3
CA00401	0	505.9	3
CA00403	0	394.9	3
CA00406	0	256.5	3
CA00407	0	247.6	3
CA00408	1	357.8	3
CA00409	0	277.7	3
CA00411	0	316.6	3
CA00452	0	470.9	3
CA01862	0	210.0	6
CA01863	0	71.3	6
CA01864	0	178.3	6
CA01865	0	181.0	6
CA01866	0	191.8	6
CA01867	0	56.4	6
CA01870	0	95.8	6
CA01871	0	198.5	6
CA01872	0	141.5	6
CA01873	0	206.6	6
CA01874	0	24.7	6
CA01876	0	22.8	6
	25	61301.6	2452.1

Table 32. Turkey Sampling Data Set - Continued

Bibliography

AMRAAM SD 5011, Revision F (May 1993). *Test Procedures PRAT Program for AMRAAM.* Contract F08626-91-C-0093.

Casella, G. and George, E. I. (1992). Explaining the Gibbs Sampler. *The Amer. Statist.* **46:3** 167–174.

Chung, K. L. (1974). *A Course in Probability Theory.* San Diego: Academic Press Inc.

Cohen, A. C. (1991). *Truncated and Censored Samples.* New York: Marcel Dekker, Inc.

Cox, D.R. (1973). Regression Models and LifeTables. *J. R. Statist. Soc. B* **34** 187–220.

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *Ann. Statist.* **7** 1–26.

Efron, B. (1981). Censored Data and the Bootstrap. *J. Amer. Statist. Assoc.* **76** 312–319.

Efron, B. and Tibshirani, R. (1993). *An Introduction to the Bootstrap.* New York: Chapman and Hall.

Gelfand, A. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *J. Amer. Statist. Assoc.* **85** 398–409.

Greenwood, M. (1926). The Natural Duration of Cancer. *Reports on Public Health and Medical Subjects* **33** 1–26.

Guglielmoni, P. AMRAAM Engineer. *Air-to Air JSPO AMRAAM IPT Eglin AFB, Fl.* Personal Interview 27–28 Oct 1994.

Hewishi, M., Robinson, A. and Tubé, G. (1990). Air-to-Air Missiles. *International Defense Review.* **28** 871–876.

Hughes Air Craft (HAC) Company (July 1990). *AMRAAM System Description Vol 9 - BIT, Support Equipment, and Containers.* Tucson, Az.

Hughes Missile System Company (HMSC) (Jan 1994). *Missile Lot 5 Production Reliability Prediction and Documentation of SPT Data Vol I and II.* Canoga Park, CA.

Kabfleisch, J. D. and Prentice R. L. (1980). *The Statistical Analysis of Failure Time Data.* New York: John Wiley and Sons.

Kaplan, E.L. and Meier, P. (1958). Non-parametric Estimation from Incomplete Observations. *J. Amer. Statist. Assoc.* **53** 457–481.

Kobren, W. Maj USAF AMRAAM Maintenance IPT Leader. *Air-to Air JSO AMRAAM IPT Eglin AFB, Fl.* Telephone Interview 2 Nov 1994.

Lagakos, S. W. (1979). General Right Censoring and Its Impact on the Analysis of Survival Data. *Biometrics* **35** 139–156.

Lawson, K. L. (1994). Bayesian Nonparametric Estimation Via Gibbs Sampling for Coherent Systems With Redundancy. Ph.D. Dissertation, Florida State University.

Miller, R. G. (1983). *Survival Analysis* New York: John Wiley and Sons.

Miller, R. G. (1983). What Price Kaplan-Meier?. *Biometrics* **39** 1077–1081.

Naval Air Warfare Center (NAWC) Weapons Division (Mar 1992). *Test Results of the AMRAAM Lot 2 Sublot 3 Hughes PRAT Program.* Point Mugu, Ca.

Nelson, W. (1972). Theory and Applications of Hazard Plotting for Censored Failure Data. *Technometrics* **14:4** 945-966.

Peto, R. and Peto, J. (1972). Asymptotically Efficient Rank Invariant Test Procedures with Discussion. *J. R. Statist. Soc. A Observations.* **135** 185–206.

Ross, S. M. (1983). *Stochastic Processes.* New York: John Wiley and Sons.

Ross, S. M. (1993). *Introduction to Probability Models.* Boston: Academic Press Inc.

Vita

Captain Denhard is a doctoral student in the Operations Research Department at the Air Force Institute of Technology. Captain Denhard was born 16 Nov 65 in New Brunswick, New Jersey. Upon completing a Bachelor of Science degree in mechanical engineering from Carnegie Mellon, Captain Denhard was commissioned into the United States Air Force through the Reserve Officer Trainning Corp program in May, 1988. Assigned to the Training System Program Office at the Aeronautical Systems Center in January, 1989, Captain Denhard managed several aircraft simulator acquistion programs including the F-15 and F-16 unit training device and the F-15, B-1, and B-2 improved visual system upgrade. In March, 1995, Captain Denhard earned a Master of Science in Operations Research at the Air Force Institute of Technology. Captain Denhard is married to the former Carol Dougherty of St Charles, Missouri. They have a son Ryan, age 2.

Permanent address: 8760 Deer Plains Way
Huber Heights, Ohio 45424

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 1995

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
ESTIMATION OF THE CAPTIVE-CARRY SURVIVAL FUNCTION FOR THE ADVANCED MEDIUM RANGE AIR-TO-AIR MISSILE (AMRAAM)

5. FUNDING NUMBERS

6. AUTHOR(S)
David R. Denhard, Capt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GOR/ENC/95M-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
ASC \ YAA
207 West D Ave Suite 622
Eglin AFB FL 32542-6844

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This thesis considers the problem of estimating the survival function of an item (probability that the item functions for a time greater than a given time t) from sampling data subject to partial right censoring (a portion of the items in the sampling data have not yet been observed to fail). Specifically, the thesis describes several parametric and non-parametric statistical models that can be used when the sampling data is subject to partial right censoring. These models are applied to the case of estimating the captive-carry survival function of the AIM-120A Advanced Medium Range Air-to-Air Missile (AMRAAM).

14. SUBJECT TERMS
AMRAAM, Reliability, Survival Analysis, Censored Data

15. NUMBER OF PAGES
202

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std Z39-18

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements.***

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	PR	- Project
G	- Grant	TA	- Task
PE	- Program Element	WU	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- DOE - See authorities.
- NASA - See Handbook NHB 2200.2.
- NTIS - Leave blank.

Block 12b. Distribution Code.

- DOD - Leave blank.
- DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- NASA - Leave blank.
- NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

*U.S.GPO:1990-0-273-271